Radio Frequency Spread Spectrum
For Distributed Control Systems

by

Ghaleb Baquir Al-Dandan

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

January, 1990

Order Number 1355710

Radio frequency spread spectrum for distributed control systems

Al-Dandan, Ghaleb Baquir, M.S.

King Fahd University of Petroleum and Minerals (Saudi Arabia), 1990

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

RADIO FREQUENCY SPREAD SPECTRUM FOR DISTRIBUTED CONTROL SYSTEMS

BY

GHALEB BAQUIR AL-DANDAN

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

JANUARY, 1990

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN, SAUDI ARABIA

COLLEGE OF GRADUATE STUDIES

This thesis, written by **GHALEB BAQUIR AL-DANDAN** under the direction of his Thesis Advisor and approved by his Thesis Committee, has been approved to and accepted by the Dean of the College of Graduate studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** in **ELECTRICAL ENGINEERING**.

<u>THESIS</u> <u>COMMITTEE</u>

Thesis Advisor, Dr. M.S. EL-Hennawey

Member, Dr. M. EL-Shafei Ahmed

Member, Dr. T.M. Bakri

Department Chairman

Dean, College of Grauate Studies

College of Graduate Studies
Office of the Dean

Date : Jan 8th 1990

THIS THESIS IS DEDICATED TO MY MOTHER, FATHER, SISTERS AND AUNT.

ACKNOWLEDGMENT

Acknowledgment is due to King Fahd University of Petroleum and Minerals for support of this research.

I would like to express my appreciation to my principal advisor, Dr. M.S. El-Hennawey, for his valuable suggestions and guidance. I am also grateful to Dr. M. EL-Shafei Ahmed, thesis committee member, whom I owe much of my conceptual understanding of the subject and who provides me with lot of support and encouragement during difficult and hard moments. I also wish to thank Dr. T.M. Al-Bakri, thesis committee member, for his useful comments and help.

I am also indebted to Dr. S.O. Duffuaa for his assistance during my network simulation. Another debt is owed to my colleagues A.S AL-Shwaiheen, S. AL-Qatri, and A.I. Ismail for their help and sincere cooperation.

Finally, I wish to thank Computer and Communication Services Department from Saudi Arabian American Oil Company (SAUDI ARAMCO) and in particular E.A. Quarterson, supervisor of Transmission and Data unit, for his support through my M.S program.

TABLE OF CONTENTS

CHAPTER 4

PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM USING NETWORK SIMULATION

CHAPTER 5

CONCLUSIONS & SUGGESTIONS FOR FURTHER STUDIES

APPENDIX-A

APPENDIX-B

اسم الطالب : غالب باقر حسين الدندن
عنوان الرساله : الإنتشار الطيفي لموجات
الراديو في انظمـة التحكم الموزعـه
التخصص : هندسـة كهربائيـه
تاريخ الدرجه : ديسمبر ١٩٨٩م

لقد كان لإستخدام طرق الإنتشار الطيفي في مجالات الإتصالات والملاحه وأجهزة الإختبار الأثر الفعال في الاونه الاخيره حيث لم يكن ممكناً الحصول علىٍ نتائج مماثله باستخدام الطرق التقليديه الأخرى .

وقد تم في هذه الرساله عرض لتصميم شبكه تحكم موزعه متكامله حيث استخدم إنتشار الطيف مباشر التتبع للإتصال بين الوحدات الطرفيه النائيه والحاسب المضيف . ولقد بني هذا التصميم الجديد على نظام التضمين المفتاحي لفارق الطور ، وكذلك نظام الشفره المقسمه متعددة التحصيل . وتم أيضاً في هذا التصميم استخدام برتوكول جديد للاستعلام الدوري بيت الحاسب المضيف والوحدات الطرفيه النائيه .

ولقد أجريت محاكاه أو تمثيل للشبكه المصممـه باستخدام مجموعه بـرامج « سلام ـ ٢ » وذلك لتقييم النظام المقترح تحت ظروف التشغيل المختلفه من حيث معدلات سرعة نقل المعلومات ونسبةالخطأ فيها . وقد شملت المحاكاه للحالتين التاليتين :

١ ـ تحصيل المعلومات فقط .

٢ ـ تحصيل المعلومات مع التحكم .

وقد أظهرت نتائج المحاكاه جدوى النظام المقترح حيث يعطى القائمين على تصميم الشبكات المرونه في اختيار معاملاتها تبعاً لإحتياجهم .

ومن المتوقع ان يساهم النظام المقترح مساهمه فعاله في تطوير سبل التشغيل القائمه حالياً في شبكات التحكم الموزعه .

درجة الماجستير في العلوم
جامعة الملك فهـد للبترول والمعادن
الظهران ـ المملكه العربيه السعوديه
ديسمبر ١٩٨٩

ABSTRACT

Ghaleb Baquir AL-Dandan

RADIO FREQUENCY SPREAD SPECTRUM FOR DISTRIBUTED CONTROL SYSTEMS

Major Field : Electrical Engineering

The application of spread spectrum techniques in recent years to communications, navigation and test equipment has produced significant results which are not possible with standard techniques. In this thesis, direct sequence radio frequency spread spectrum (RF/SS) is used for communication between Remote field Terminal Units (RTUs) and host computer in Distributed Control System (DCS) applications. This new system is based on differential phase shift keying modulation and code-division multiple-access. It also uses a new polling access protocol between the host computer and the RTUs.

Overall system is simulated, using a SLAM II software package, for different data rates and various bit-error probabilities under two separate cases : 1) Data acquisition and 2) Data acquisition and control. Network simulation indicates the feasibility of the proposed access protocol and provides the system designer with variety of options to select the network parameters based on his requirements.

The proposed system is expected to yield substantial improvement in the operation of present DCS systems.

MASTER OF SCIENCE
KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN, SAUDI ARABIA

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

AMDT : Average Message Delay Time

ATDMA : Asynchronous Time Division Multiple Access

BC : Buffer Capacity

BPSK : Binary Phase Shift Keying

BRAM : Broadcast Recognition Access Method

BW : Bandwidth

C : Code Spreading Sequence

CCS : Centralized Control System

CDMA : Code Division Multiple Access

CF : Control Frame

CNC : Computer Numeric Control

CPU : Central Processing Unit

CRC : Cyclic Redundancy Check

CSMA : Carrier Sense Multiple Access

DB : Decibel

DCS : Distributed Control System

DISC : Disconnect

DPSK : Differential Phase Shift Keying

DQAM : Differential Quadrature Amplitude Modulation

DQPSK : Differential Quadrature Phase Shift Keying

DS : Direct Sequence

E_b : Energy per Bit

ED : End Delimiter

F : Total walk time

FC : Frame Control

FCC : Federal Communication Commission

FCS : Frame Check Sequence

FDDI : Fiber Distribution Data Interface

FDMA : Frequency Division Multiple Access

FH : Frequency Hopping

G(f) : Fourier Transform

G_p : Process Gain

HDLC : High Level Data Link Control

IEC : International Electrotechnical Commission

IEEE : Institute of Electrical and Electronics Engineers

I.F : Information Frame

IF : Intermediate Frequency

I/O : Input / Output

ISDN : Integrated Services Digital Network

ISM : Industrial, Scientific and Medical

ISO : International Standard Organization

J : Number of Interferers

L : Internal System Losses

LLC : Logical Link Control

LPF : Low Pass Filter

m	:	Number of Field Points connected to a particular RTU
MACS	:	Mixed ALOHA Carrier Sense
MAC	:	Medium Access Control
MAP	:	Manufacturing Automation Protocol
MIL	:	Military
M_j	:	Jamming Margin
MSRG	:	Modular Shift Register
n	:	Number of RTUs
N	:	Number of Frequencies
N_o	:	Noise Power Spectral Density
NRM	:	Normal Response Mode
OSI	:	Open System Interconnection
p	:	Number of Shift Registers in the linear code generator
Pe	:	Probability of error
PLC	:	Programmable Logic Controller
PROFI	:	Process Field (PROFIBUS = Process Field Bus)
R	:	RF Bandwidth
RF	:	Radio Frequency
RTN	:	Number of RTUs
RTU	:	Remote Terminal Unit
RXr	:	Receiver
S	:	Modem synchronization & overheads delay
SA	:	Station Address

SAW : Surface Acoustic Wave

SD : Start Delimiter

SF : Supervisory Frame

S(f) : Power Spectral Density

SIM : Set Initialization

SIT : System Idle Time

SLAM : Simulation Language Alternate Modeling

S/N : Signal to Noise ratio

SP : Data Rate

SPF : Special Purpose Frame

SS : Spread Spectrum

STD : Standard

STDMA : Statistical Time Division Multiple Access

T_c : Code sequence pulse duration

T_i : Information rate pulse duration

T_m : Processing Time

T_p : Polling Message Delay

T_s : Scan time

T_t : Transmission Time

TH : Time Hopping

TXr : Transmitter

u : Frequency Swept rate

w : Angular Velocity

w : Walk-time

XID : Exchange Identification

Y : Round trip propagation delay

CHAPTER 1

INTRODUCTION

1.1 GENERAL

Supervising various plant system applications in a fast, accurate and reliable manner is a major concern of the industry (e.g. : oil, gas, steel etc..).

Centralized and distributed control systems are two different system architectures applied to monitor the plant activities. In centralized control systems (CCS), a single computer called " host " monitors and controls the plant. If the main computer fails, the whole system is affected. On the other hand, distributed control systems (DCS) is based on multi-host system architecture. In such systems, the processing power and data bases are distributed at various distinct locations.

Some properties of the DCSs are :

1. Decentralize system work load : the central computer is relieved from being the only processing unit.

2. Fast response time : jobs will be executed in a shorter time since two or more CPUs are available.

3. Reduce communication costs : many jobs can be processed locally without the need for a central computer, thus reducing the channel cost/ service.

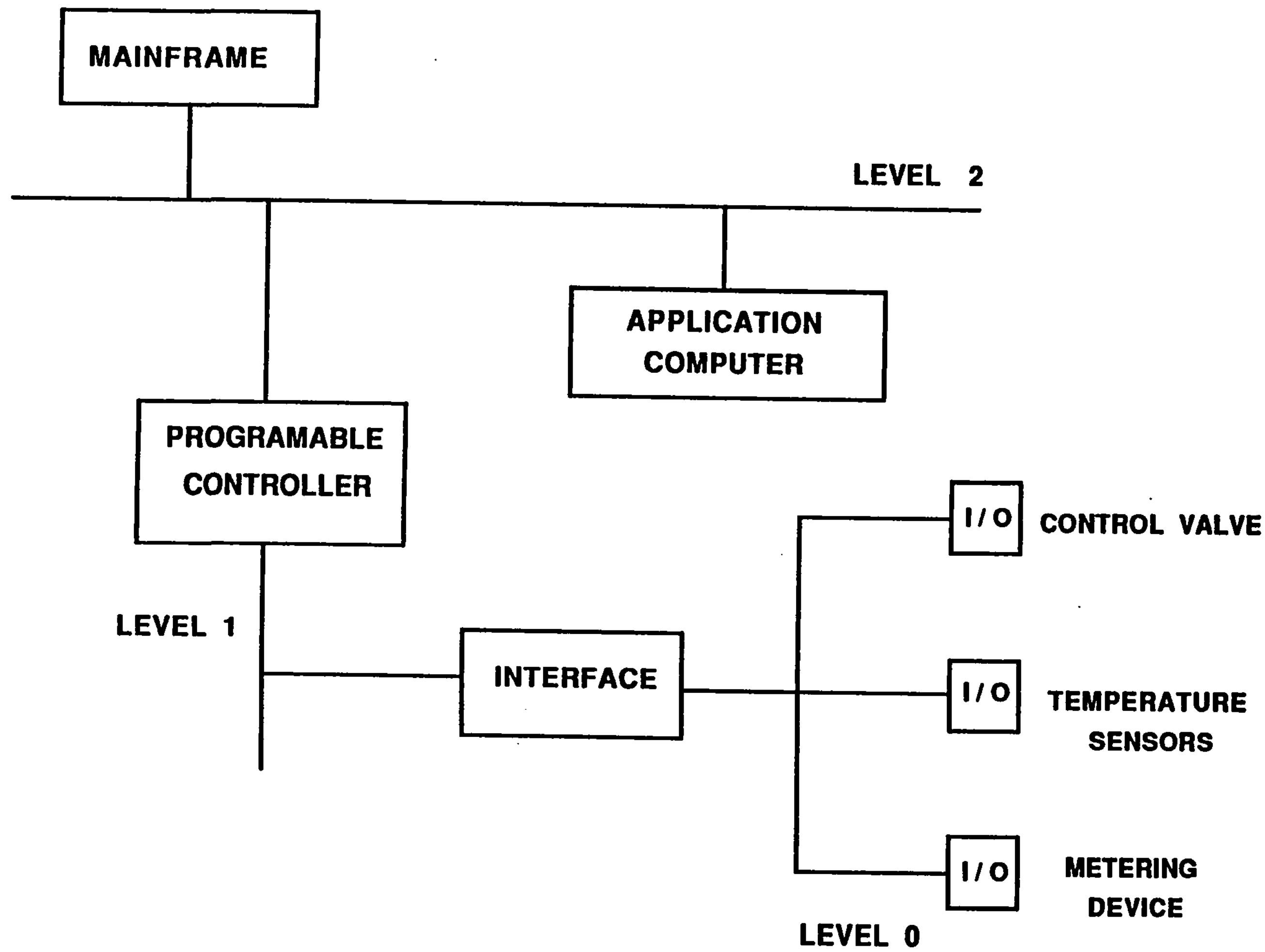
4. Increase system reliability : losing one CPU will not affect the performance of the other computers.

In the following sections a brief idea about some problems faced by current industries are discussed, then an overview of the present solutions taken to overcome such problems is presented, and finally a new solution is proposed based on the Manufacturing Automation Protocol (MAP).

1.2 PROBLEM FORMULATION

Factory communications are generally divided into three levels as shown in Figure 1.1 [1] :

o Level 0 (zero) links the field points (sensors, actuators,etc..) to the secondary stations. This is



DISTRIBUTED CONTROL SYSTEM

FIGURE # 1.1

usually accomplished by point-to-point cable connections.

- Level 1 (one) corresponds to the secondary stations such as Programmable Logic Controllers (PLCs), robots and Computer Numerical Control (CNC) machines .

- Level 2 (two) is responsible for the management of the whole system. It interconnects various application computers.

Level 0 (zero) of the existing distributed control systems suffer from excessive field cable wiring. Field points (sensors, actuators) have to be connected to Programmable Logic Controllers (PLCs) through cable facilities. Furthermore, installation and maintenance costs due to changes (e.g.: addition of control points) cannot be easily achieved.

1.3 PRESENT SOLUTION

Presently the answer to the increasing complexity of cable wiring at level 0 (zero) is through the application of the Field Bus concept. Field buses are designed to replace the point-to-point links (from each sensor to its controlling equipment), by a single link on which all information is transmitted serially and multiplexed in time.

The draft specification of the Field Bus Standard [2] was announced by the International Electrotechnical Commission (IEC) as follows :

1. The standard should define a serial digital communication link to and from field devices.

2. Part of the standard must support power and data over the bus.

3. Open System Interconnection (OSI) technology should be used where possible. Data formats should be specified for lower levels.

4. Twisted pair , coaxial cable and fiber optic connections

will be supported. <u>Radio links may be supported in the future</u> .

5. Typical response time performance has been specified at 20ms whilst a high performance option has been specified at 5ms .

6. The Bus length, number of addressable devices per bus and the data rate must be optimized.

Some practical implementations which meet the IEC for a Field Bus are, the MIL-STD. 1553 [3], and the FIP [4]. Other proposals such as BITBUS [5], PROWAY C [6], and PROFIBUS [7] are considered to be low end solutions that can handle process control applications with a medium response time.

Currently a great effort is being carried out to develop the "International Field Bus Standard" with the following objectives in mind :

1. Reduced installation costs via better wiring topologies.

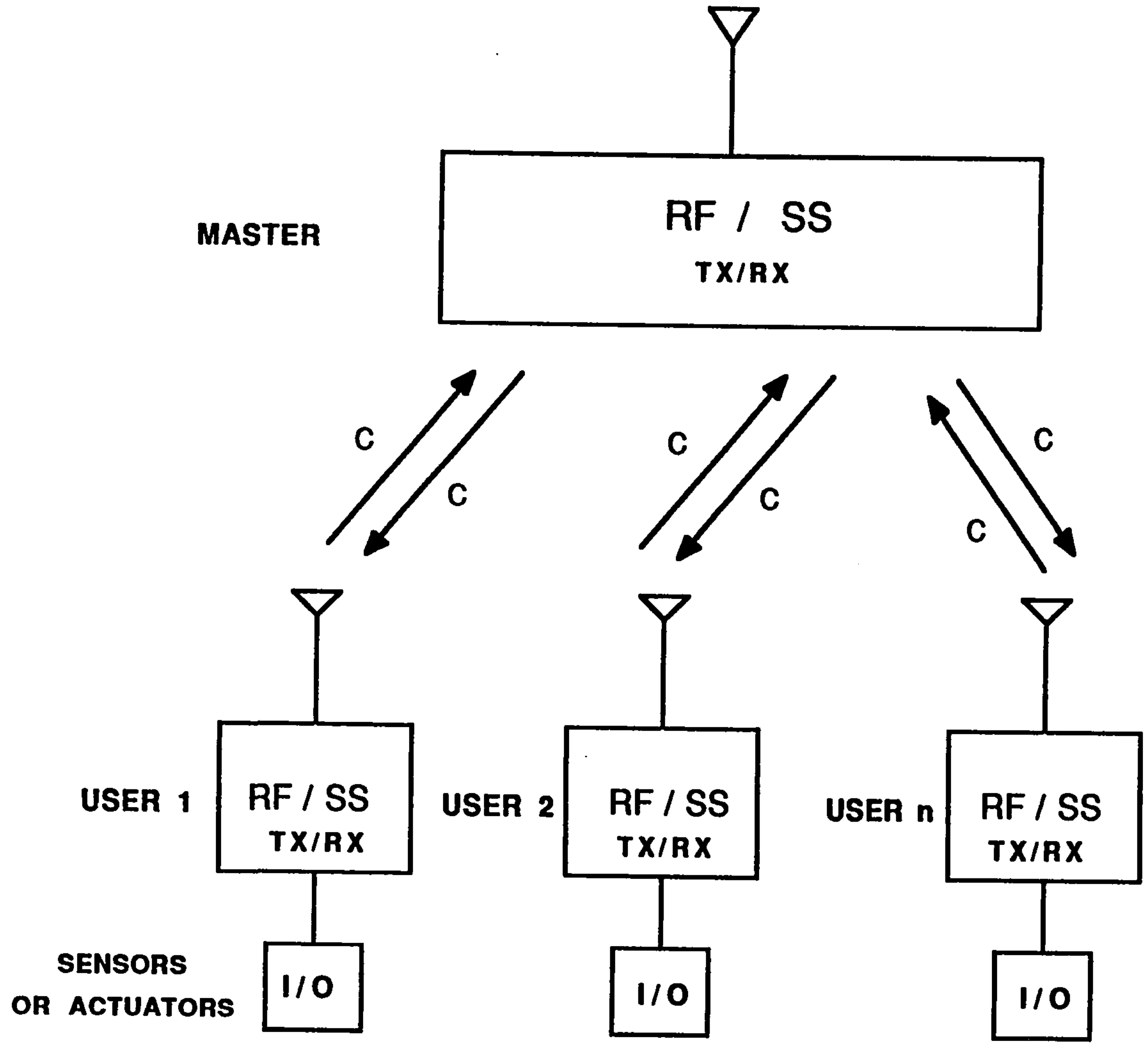
2. System flexibility: system changes can be easily performed.

3. Improved data security.

This solution of running the single cable through the entire plant will reduce some of the existing complexities in cable wiring [7]. The major disadvantage of such technique is the reduced system reliability since a damage in the single cable will affect the overall system. Furthermore, the system is not secure enough to prevent people from tapping into the cable.

1.4 PROPOSED TECHNIQUE

In this thesis, we propose a new method by which we will apply direct sequence radio frequency (RF) spread spectrum technique for communication between Remote Terminal Units (RTUs) at level 0 (zero) and Programmable Controllers at level 1 (one). Figure 1.2 shows " n " different users (RTUs) communicating with a master station through an RF/ spread spectrum link. Every RTU has its own address and is connected to " m " field points. The master station polls each RTU separately and uses a code pattern C as its code spreading sequence. The receiver circuitry of the RTUs are matched to C, and the addressed RTU will only respond to the



PROPOSED RF SS SYSTEM

FIGURE 1.2

polling message. The master station receiver is also matched to C in order to accept the RTU transmitted sequence.

1.5 PERFORMANCE EVALUATION

The major issues involved in the evaluation of the proposed system are :

1. The type of RF spread spectrum technique to be used.
2. The access method and procedure to be adopted.
3. The code sequence generator required.
4. The Transmitter / Receiver circuitry to be designed.
5. The overall network simulation.

In Chapter 2 we review RF spread spectrum systems and the various multi-access techniques. Chapter 3 is devoted to the operation and design of the proposed system. In Chapter 4 we evaluate the proposed system using network simulation. Finally, in Chapter 5 we present the conclusions to our research.

In the next chapter, the reasons behind the choice of spread spectrum communication system are explained, different types of access techniques used for the communication between the master station and the RTUs are also evaluated.

CHAPTER 2

LITERATURE SURVEY of SPREAD SPECTRUM SYSTEMS and MULTI-ACCESS TECHNIQUES

In this chapter, the concept of spread spectrum communication systems, its types, properties, and code sequence generators are presented. Furthermore, different access techniques used in computer networks are discussed and only one technique, namely the Roll-Call Polling is found to be highly applicable in our proposed RF spread spectrum system.

2.1 SPREAD SPECTRUM SYSTEMS

Spread Spectrum is the art of expanding the bandwidth of a signal, transmitting that expanded signal, and recovering the desired original information bandwidth. Therefore, a spread spectrum system must meet two important criteria :

1. The transmitted bandwidth is much greater than the bandwidth of the information being sent.

2. The information signal must be added to another function, such as the spectrum - spreading code before being transmitted.

In the past decade, spread spectrum technology was restricted for military communications; where it is attractive for its resistance to interference, security and capability of high resolution ranging.

The application of spread spectrum techniques in recent years to communications , navigation and test equipment have produced significant results which are not possible with the standard techniques.

Some properties of spread spectrum systems are :

1. Selective Addressing :

If the code sequence generator of a particular receiver is known, a spread spectrum transmitter with the same code sequence generator can send messages to that receiver [8].

2. Code Division Multiplexing :

A number of transmitters having different code sequence generators can transmit at the same frequency and time, with minimal interference from one transmitter to another, and with complete recovery of the signals by their intended receivers.

3. Low-Density Power Spectrum :

The power transmitted by a spread spectrum system is spread over a wide band when compared with conventional systems. Thus the power per Hertz (power density) is lower, and this property leads to the fact that spread spectrum systems are being called as "low probability of intercept" systems.

4. Security :

The code sequence generators used have the capability of preventing unauthorized users from accessing the information.

5. Multiple Rejection (Multi-path Interference) :

Spread spectrum systems can reject multi-path signals whose delay is greater than the reciprocal of the code rate.

Due to those properties the United States' Federal Communication Commission (FCC) has assigned several bands for commercial applications of spread spectrum systems [9]. The frequency authorization includes 902-928 MHZ, 2400-2500 MHZ and 5725-5850 MHZ bands, as long as the spread spectrum systems do not interfere with other licensees in these spectra.

Spread Spectrum is currently used in packet radio [10,11]. It has been suggested for mobile radio [12], has been experimented for wireless terminals [13], and wireless PABXs [14]. More recently, spread spectrum has been proposed for use in the office environment [15,16,17], small local area network using power lines [18], and in power distribution remote measurement [19] .

Spread spectrum techniques can be classified into four different types, Direct Sequence (DS), Frequency Hopping

(FH), Time Hopping (TH), and Chirp Systems. Each type has a useful application and various hybrid combinations of such types are also possible.

2.1.1 Direct Sequence (DS)

DS spread spectrum is the modulation of the carrier frequency by a special modulating signal . The modulating signal is a digital stream of bits which consists of a code sequence, mod-2 added with the digital information sequence. The important parameters in DS/SS systems are the process gain and the *jamming margin*.

Process Gain (G_p):

G_p is the difference in signal-to-noise ratio between the output and the input signals . In spread spectrum systems the process gain is defined as :

$$G_p = \frac{\text{Transmitted RF Bandwidth}}{\text{Information Bandwidth}}$$

$$G_p = \frac{R \text{ (RF BW)}}{I \text{ (INF BW)}} = \text{Code Length} \qquad (2.1.1)$$

Another general method to express the processing gain is through the power spectrum.

Assume the followings :

T_i= Pulse duration of the information sequence I(t).

T_c= Pulse duration of the code sequence C(t).

G(f) = Fourier transform of the rectangular pulse T_i.

$S_i(f)$= Power spectral density of I(t).

$S_c(f)$= Power spectral density of C(t).

In general, we can approximate any rectangular pulse x(t) of duration T and amplitude A by the following Fourier transform X(f) :

$$X(f) = \int_{-W}^{+W} A \exp(-j2\pi ft)dt \qquad (2.1.2)$$

Therefore,

$$G(f) = \int_{-Ti/2}^{+Ti/2} A \exp(-j2\pi ft)\,dt$$

$$= AT_i \sin(\pi fT_i)/\pi fT_i$$

$$= AT_i \operatorname{sinc}(fT_i) \qquad (2.1.3)$$

This expression is the voltage distribution for the signal whose power spectrum is :

$$S(f) = |\ G(f)\ |^2$$

$$S_i(f) = A^2 * T^2{}_i * |\operatorname{sinc}^2(fT_i)|^2 \qquad (2.1.4)$$

Similarly, the power distribution for the coded sequence is:

$$S_c(f) = A^2 * T^2{}_c * |\operatorname{sinc}^2(fT_c)|^2$$

Let J(t) be the Jammer's signal. Thus the received signal can be expressed as :

$$r(t) = I(t) * C(t) + J(t) \qquad (2.1.5)$$

The received signal is then multiplied by the code sequence C(t). Therefore, we get :

$$y(t) = I(t)*C^2(t) + J(t)*C(t)$$

$$y(t) = I(t)+J(t)*C(t) \tag{2.1.6}$$

From equation 2.1.6 we notice that the fraction of the power due to the jammer which can pass through the filter is roughly equal to $1/(f_c T_i)$. Thus, the data have a power advantage of $f_c T_i$, the processing gain.

$$G_p = f_c T_i \tag{2.1.7}$$

Therefore, in order to increase the process gain we can either increase the RF BW or decrease the information rate transmitted.

Jamming Margin (M_j)

The jamming margin is the capability of the receiver to perform correctly in the presence of an interfering signal. The Jamming Margin is given by Dixon [20] as :

$$M_j = G_p - [L_{sys} + (S / N)_{out}] \qquad (2.1.8)$$

where,

L sys = internal system losses

(S /N) out =Signal-to-Noise ratio at the information output

NOTE : $M_j < G_p$ (2.1.9)

Different modulation schemes could be used in the direct sequence spread spectrum systems . The most commonly used technique is the Binary Phase Shift Keying (BPSK). In BPSK the carrier signal is transmitted with a certain phase corresponding to the input binary digit 1 , and with 180^o phase difference corresponding to binary digit 0 .

Balanced modulators are often used in BPSK systems for the following reasons :

1. It suppresses the carrier component thus helping to make the signal more secure .

2. No power is wasted in transmitting the carrier.

2.1.2 Frequency Hopping (FH) :

Frequency Hopping can be defined as: " A multiple frequency, code selected, Frequency Shift Keying" [20]. A practical frequency hopping spread spectrum system would have thousands of frequencies available for use. Each frequency is chosen on the basis of the input code sequence in combination with the information sequence. The number of the frequencies used with the input data rate determines the RF bandwidth, and process gain and jamming margin.

Process Gain (Gp) :

The process gain for non-contiguous frequency hopping system is :

$$G_p = \text{Number of available frequencies} = N$$

$$G_p \text{ (dB)} = 10 \text{ LOG (} N \text{)} \qquad (2.1.10)$$

This process gain doesn't consider any crosstalk (or inter-channel interference). The expected error rate for "J" interferers is = J/N.

The major disadvantages in using frequency hopping systems as compared to DS/SS systems are:

1. High bandwidth requirement

2. A stable frequency synthesizer is needed

3. Transmitter power should be identical in every channel

2.1.3 **Time Hopping (TH) :**

The input data is used to key a transmitter " on " when a one is transmitted and " off " when a zero is transmitted. This technique has the advantage of reduced transmit power on the average of 50 % but it suffers from a low interference rejection because a continuous carrier signal at the center frequency can block communication. Therefore, this technique is used mainly in combination with other schemes of SS.

2.1.4 **Chirp Systems**

This type of spread spectrum systems does not contain any code sequence to generate the processing gain but rather employs a frequency-swept in the transmitted signal. In order to develop the transmitted signal, a linear voltage sweep is applied to a voltage-controlled oscillator. Thus, the RF transmitted signal frequency varies in a known way according to the frequency - swept rate :

$$u = dw/ dt \tag{2.1.11}$$

w = angular frequency

The transmitted signal F(t) [20], is given as :

$$F(t) = A \cos (w_c t + 0.5 * u * t) \tag{2.1.12}$$

This technique has found its major application in radar systems [20].

Based on the above discussion the direct sequence spread spectrum will be considered as will be seen in the next chapter, where the design of both transmitter and receiver circuitry are discussed. In the next section, a brief discussion about the various code sequence generators used in RF/SS systems is presented.

2.2 CODE GENERATORS

Coding techniques play an important role in all digital communication systems. The code sequence of interest to us is that intended for bandwidth spreading and not for the direct transfer of information.

The following properties are common to all codes for use in spread spectrum systems :

1. Interference Protection : Coding techniques improve the signal to interference ratio by increasing the process gain of the system.

2. Privacy : Coding techniques help to hide the information being transmitted, thus providing system security.

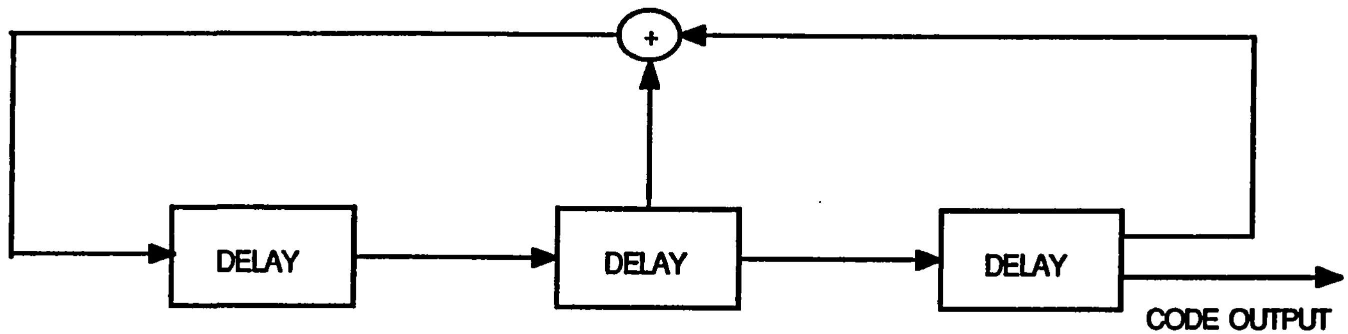
2.2.1 Linear Code Generators

The simplicity of deciphering linear codes makes their usage suitable for application in spread spectrum systems. A linear code generator is made up of delay elements (shift registers) in conjunction with feedbacks. Fig. 2.1 illustrates a typical example of linear generators. The polynomial expressing the code generator can be expressed as follows :

$$f(x) = 1x^{o} + A_1x^1 + A_2x^2 + \text{etc.} \ldots + A_px^p$$
(2.4.1)

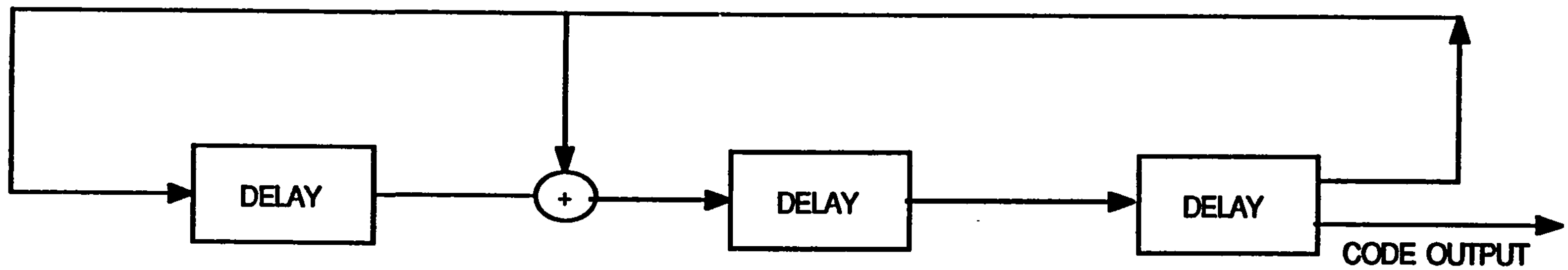
where p is the number of shift registers used and A_p takes the value of either zero (no feedback) or one (with feedback).

A much preferable and an equivalent sequence generator configuration to fig 2.1 is the modular type of linear code generators. In such technique the same code sequence can be generated with the same hardware requirement. Fig 2.2 illustrates Modular Shift Register Generators (MSRG) in which the mod-2 feedback adders are placed between stages. This configuration reduces the total delay and ensures that it is less than the code rate period [20].



LINEAR CODE GENERATOR

FIGURE # 2.1



MODULAR LINEAR CODE GENERATOR

FIGURE # 2.2

2.2.2 Maximal Code Sequences

Maximal Code Sequences are the longest number of codes that can be generated by a given code generator. For any p-stage shift registers, there are $Q(2^p - 1) / p$ maximal sequences, where $Q(x)$ is an Euler number - the number of positive integers including 1 that are relatively prime to and less than x .

Some properties of the maximal code sequences are :

1. The total number of ones in a given code generator always exceeds the total number of zero's by one.

$$\text{Total number of ones} = 2^p / 2 \tag{2.4.2}$$

$$\text{Total number of zero's} = (2^p / 2) - 1 \tag{2.4.3}$$

Where p is the number of shift registers used in the code generator.

2. The addition of two maximal sequences, each of length r, would produce r different r-chip nonmaximal linear codes in addition to the two basic linear maximal codes.

3. White [21] and Tansworthe [22] showed that if randomness in the maximal code sequence is desired, the code length should be increased.

4. The statistical distribution of ones and zeros is well defined.

5. The cross-correlation function for maximal length sequences of the same length L was shown by Anderson [23] to be :

$$\left| \Phi(t) \right| \leq \left(\frac{1 + 1/L - 1/L^2}{L} \right)^{1/2} \qquad (2.4.4)$$

As L goes to ∞

$$\left| \Phi(t) \right| \text{ goes to } 1/(L)^{1/2} \qquad (2.4.5)$$

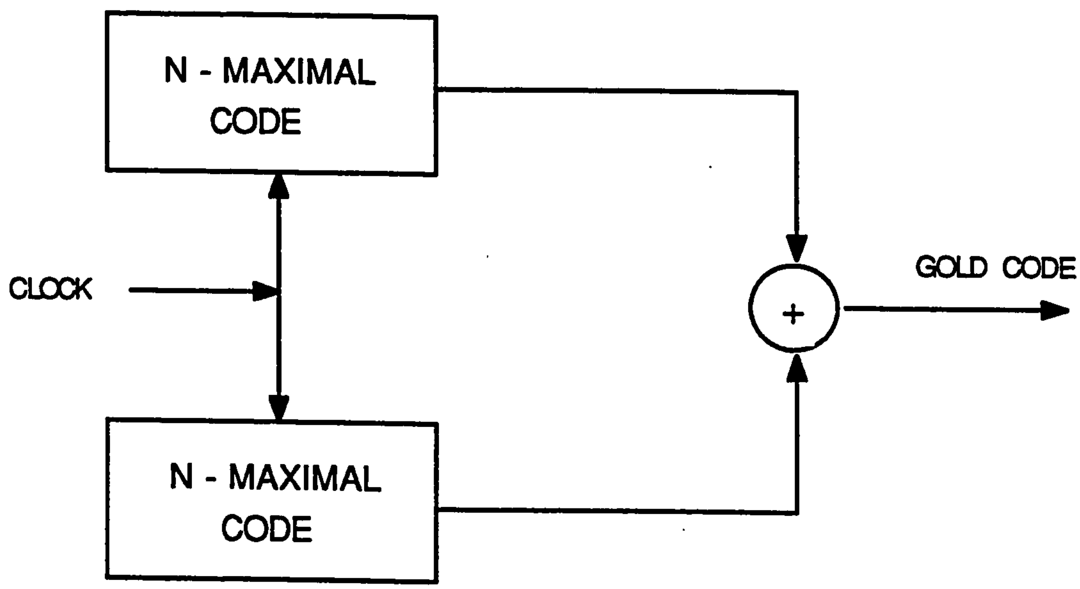
2.2.3 Gold Code Sequence Generators

Gold codes [24,25] belong to the general class of composite codes. Gold codes are generated by the mod-2 addition of two linear maximal codes as shown in figure 2.3. These codes are very useful for generating large number of codes. The total number of maximal length sequences that could be provided by Gold codes are equal to $(2^P - 1) + 2$ [20].

In this thesis, Gold codes are chosen since they provide a bounded and a uniform cross correlation between different code sequences. Anderson expression for the cross correlation bound is :

$$|\bar{\Phi}(t)|_G \le \left(\frac{(2)^{1/2} * (1+1/L)^{1/2} + 1/(2)^{1/2}}{(L)^{1/2}} \right)^{1/2} \qquad (2.4.6)$$

As L goes to ∞ , then :



GOLD CODE GENERATOR

FIGURE # 2.3

$$| \bar{\Phi}(t) |_G \leq \frac{(1 + (2)^{1/2})}{(L)^{1/2}} \qquad (2.4.7)$$

Therefore,

$$| \bar{\Phi}(t) |_G = (2)^{1/2} * | \bar{\Phi}(t) | \qquad (2.4.8)$$

From equation (2.4.8) we deduce that maximal codes exhibits a cross correlation of $(2)^{1/2}$ advantage over Gold codes of the same code length L.

In the next section, we will discuss the various access techniques that are widely implemented in computer networks.

2.3 MULTI-ACCESS TECHNIQUES

Multi-access protocols arise from the need of independent users to share an expensive resource . Users can gain access to a communication channel by adopting a common access method. Multi-access techniques can be divided into four major categories:

1. Fixed Access Assignment
2. Random Access Assignment
3. Adaptive and Mixed mode access Assignments
4. Demand Access Assignment

2.3.1 FIXED ACCESS ASSIGNMENT

A fixed allocation of a channel resource (time or frequency) is assigned for different users in a predetermined manner. There are two major types of fixed assignment :

1. Frequency Division Multiple Access (FDMA): It consists of assigning a fraction of the overall bandwidth to every user.

It can be shown that if the channel output rate is equal to Z bits/sec and the average message length = 1/u bits/packet, then for M users the expected average packet delay time is [26]:

$$E(T)_{FDM} = \frac{M}{u*Z} + \frac{M*\underline{n}}{2(1-\underline{n})*u*Z} \qquad (2.3.1)$$

where $\underline{n}$ is the traffic intensity equal to :

$$\underline{n} = \frac{M*s}{u*Z} \qquad (2.3.2)$$

and s is message arrival rate (messages/sec) per user.

2. Time Division Multiple Access (TDMA): It can be divided into two parts. Synchronous Time Division Multiple Access (STDMA) and Asynchronous Time Division Multiple Access (ATDMA).

- **STDMA** : It consists of assigning a fixed predetermined time slots to every user.

- **ATDMA** : It allocates the channel capacity for the various users in a dynamic manner and according to their demands. The multiplexer, which is connected to every user, scans the line buffers for each terminal. It switches from one terminal to another if the former is idle.

For TDM systems it can be shown that the expected delay time is [26]:

$$E(T)_{TDM} = \frac{1}{u*Z} + \frac{M}{2*u*Z} + \frac{M*\underline{n}}{2(1-\underline{n})*u*Z} \qquad (2.3.3)$$

Therefore, comparing equations (2.3.1) and (2.3.3) we get :

$$E(T)_{FDM} = E(T)_{TDM} - \frac{1}{u*Z} + \frac{M}{2*u*Z} \qquad (2.3.4)$$

Both FDMA and STDMA suffer from inefficient use of bandwidth. Some users might have no data to transmit whereas other users might be heavily loaded, yet all have a fixed bandwidth whether used or not .

At light traffic intensity ATDMA is much superior to both STDMA and FDMA . A major disadvantage for using ATDMA is that at high traffic intensity the output buffer of the multiplexer builds up easily and an overflow is highly expected .

2.3.2 RANDOM ACCESS TECHNIQUES

These techniques allow different users to access the communication channel in a random way. Some examples of such techniques are Pure Aloha [27], Slotted Aloha [28] and Carrier Sense Multiple Access (CSMA) [29]. All Random access techniques are well-suited for bursty traffic. A major draw - back of these access methods are the randomness of the response time due to the possibility of message collision. Furthermore, as the message arrival rate into the system increases, the delay time increases significantly as compared to other access methods.

2.3.3 ADAPTIVE AND MIXED MODE ASSIGNMENT

Adaptive techniques have not been generally adopted in practice . Each station on the network requires a prior knowledge of the current network load. When the load on the network reaches a certain value, the access technique changes in order to achieve better performance. Examples are Tree Walk [30], URN [31] and BRAM [32].

Mixed mode access techniques are usually a combination of two or more access methods. An example of such access techniques is Mixed ALOHA Carrier Sense (MACS).

Adaptive and Mixed mode access techniques are expensive to build since the hardware and software required to implement these techniques are not " off the shelf " .

2.3.4 DEMAND ACCESS ASSIGNMENT

There are two classes of demand access assignment :

1. Distributed Controlled Demand Assignment : The system is not dependent on a specific central processor . The users

can transmit to each other in sequential manner . Examples of such techniques are Token Passing in a bus or ring and register insertion. These methods are efficient because the network performance does not degrade rapidly as the load into the system increases.

The major disadvantages of these access techniques are :

- The operation of the network depends on each station's network interface, so if an interface fails the network is essentially broken.
- The network must be broken in order to add or delete stations.

2. Centrally Controlled Demand Assignment: In this type the master station controls the overall network . Users are only allowed to transmit when they receive a poll from the master station. This technique is generally referred to as " POLLING " [33,34].

There are two types of polling:

- Hub polling: The central processor polls the furthest station, this station then transmits its data to the central processor and signals (tells) the next station to begin transmitting etc...When all stations transmit their data the central processor regains control over the network.

- o Roll-Call polling: The central processor polls each terminal in turn. When the polling message arrives each user transmits his data and the central processor then polls the next station.

Before analyzing both polling techniques, we would like to define the scan and the walk times.

Scan Time (T_s) :

Scan time is the time between successive polls or scans of a specific station. This time is generally a random variable depending on how much data is received from each station. In this thesis we will assume this time to be fixed.

Walk Time (w) :

Walk time is the fixed portion of the scan time that attributes to propagation delay, modem synchronization and any additional message overheads.

The value of the walk time depends on the polling scheme used.

Assume the followings :

F = Total system walk time

S = Modem synchronization time + overheads

Y = Round trip propagation delay

T_p= polling message delay

$$F_{hub} = N*S + Y \tag{2.3.5}$$

$$F_{roll} = N*S + N*T_p + Y' \tag{2.3.6}$$

where Y' is the overall propagation delay which depends on the geometry of the system. That is, if the host computer is placed at the center and the terminals are spaced equally apart , then :

$$Y' = 2*(Y/N + 2Y/N + \text{etc.} \; .... + NY/(2N)) \tag{2.3.7}$$

Now if the terminals are equally spaced from one another and from the host, where the system appears to be in a multi-point configuration , then :

$$Y' = (Y/N + 2Y/N + etc. \ldots + NY/N) \qquad (2.3.8)$$

$$= (Y*(1+N))/2 \qquad (2.3.9)$$

In all cases the hub polling has the advantage of reduced time delay since the central processor does not need to send its polling message to every station. On the other hand, the hub polling requires additional hardware equipment at each terminal in order to listen to the polling message from its neighboring station. Therefore, when one station is down the whole network is affected.

In this thesis, the Roll-Call polling is chosen since the overall system will not be affected when one or more stations are down.

In the next section, we will present a brief discussion about the Manufacturing Automation Protocol (MAP). MAP details the technical requirements for a factory local area network to support communication among computers and other intelligent devices.

2.4 MAP SPECIFICATIONS

In order for the new proposed solution to be compatible with the existing plant-floor computer devices, which are supplied by different vendors, it is recommended to follow the Manufacturing Automation Protocol (MAP) specifications. MAP architecture is based on " International Standard Organization" (ISO) reference model for Open System Interconnection (OSI). The physical layer (layer 1) of the MAP is similar to IEEE 802.4 Token Passing in a bus. In modern control systems, MAP is used at levels 1 (one) and 2 (two) .

The following is a summary of the MAP specifications at layer one and two of the OSI model [35] :

- Physical layer (layer one) : broadband coaxial cable . The electrical and mechanical characteristics of the cable is given by IEEE 802.4

- Data link layer (layer two) : it consists of two sublayers the medium access control (MAC) and the logical link control (LLC).

MAC : Token passing on a bus configuration (IEEE 802.4) has been chosen as the media access control.

LLC : The IEEE 802.2 specifications are the preferred standard for the logical link control sublayer. There are three types of services within this sublayer,unacknowledged connectionless-oriented (type 1), connection oriented (type 2) and acknowledged connectionless-oriented (type 3).

MAP recommends unacknowledged connectionless-oriented (type1) to be used for the LLC sublayer.

SUMMARY

In this chapter, we have noticed that RF/SS systems offer major advantages over conventional communication systems. We have also concluded that Direct Sequence modulation is very suitable for our application due to its simplicity and small bandwidth requirement as compared to Frequency Hopping. Furthermore, the Roll - Call Polling has been recommended as the access method between the master station and the RTUs and

Gold code sequence generator has been chosen in our system design.

CHAPTER 3

THE PROPOSED SYSTEM : DESIGN & OPERATION

In this chapter, the proposed system design parameters such as RTU buffer capacity, transmission speed and probability of error are presented. Then an RF spread spectrum transmitter and receiver circuits are discussed, and finally the new access procedure adopted for the communication between master station and RTUs is explained, followed by a detailed discussion about the proposed frame format.

3.1 DESIGN PARAMETERS

Various design parameters are assumed in the network simulation programs shown in Appendix B and C. The followings are description of the main parameters considered :

1. RTU buffer capacity (BC): Each RTU buffer is allowed to have a limited number of data messages which if exceeded, that particular RTU is removed from the

polling cycle.

2. System Idle Time (SIT) : The time which the master station should wait before starting the next polling cycle. This variable is found during the simulation .

3. Data Rate (SP) : The data rate used in the simulation is based on the FCC recommendation for RF spread spectrum systems discussed in Table 3.1. Two information rates are considered, 445.63 Kbits/sec and 894.77 Kbits/sec.

4. Average RTU Message Delay Time (AMDT) : The average time that the RTU data message should wait in the buffer before it is transmitted to the master station.

5. Probability of error(P_e): Different RTU packets and polling messages error rates are considered.

6. Number of RTUs (RTN) : This variable is used as an input to the simulation program.

7. The master station polling message is designed to be ten bytes long in case of data acquisition, and thirty two bytes when control commands are activated by the host.

8. The RTU data message is set to be sixty four bytes long.

9. The polling cycle is assumed to be 100 msec.

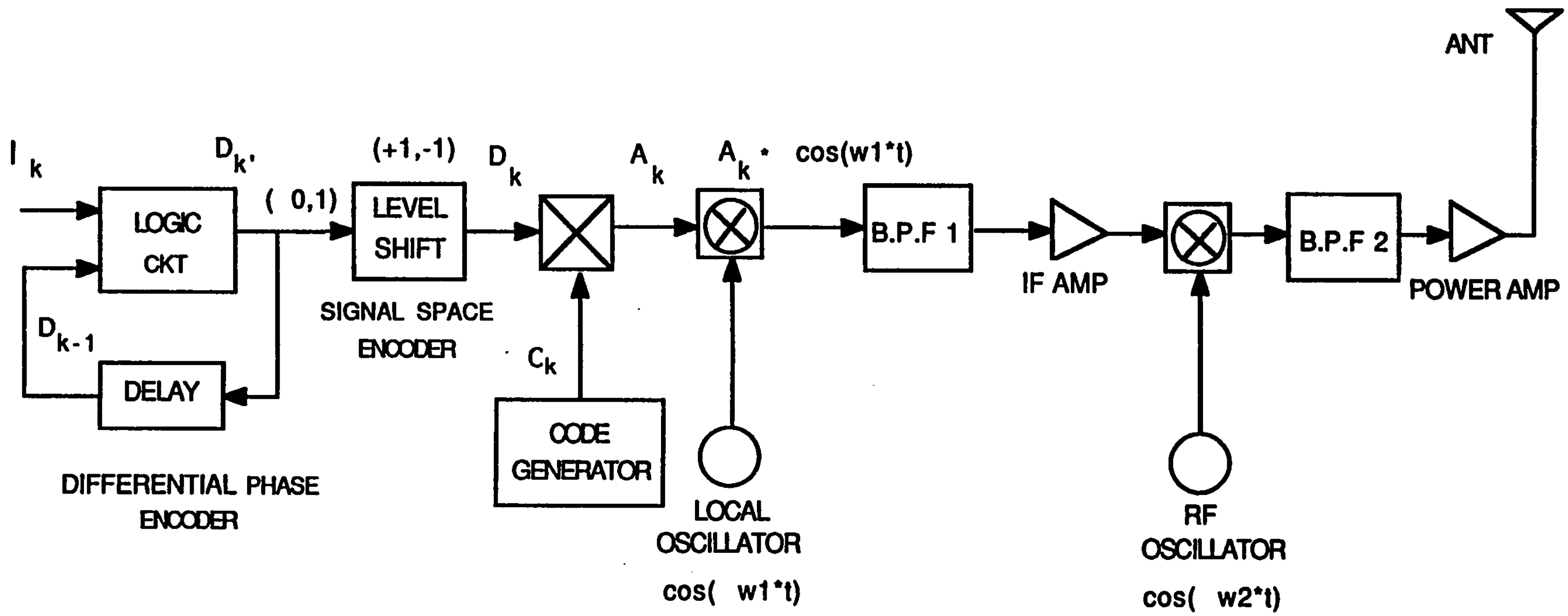
3.2 TRANSMITTER DESIGN

The transmitter design is based on direct sequence spread spectrum system using differential phase shift keying (DPSK) modulation (refer to section 3.2.1). The transmitter block diagram [36] is shown in Figure 3.1. The information bits I_k are differentially encoded using the digital logic circuit shown in Figure 3.2. The output sequence $D_{k'}$ (0 or 1) :

$$D_{k'} = I_k \cdot D_{k-1} + I_k \cdot D_{k-1} \qquad (3.3.1)$$

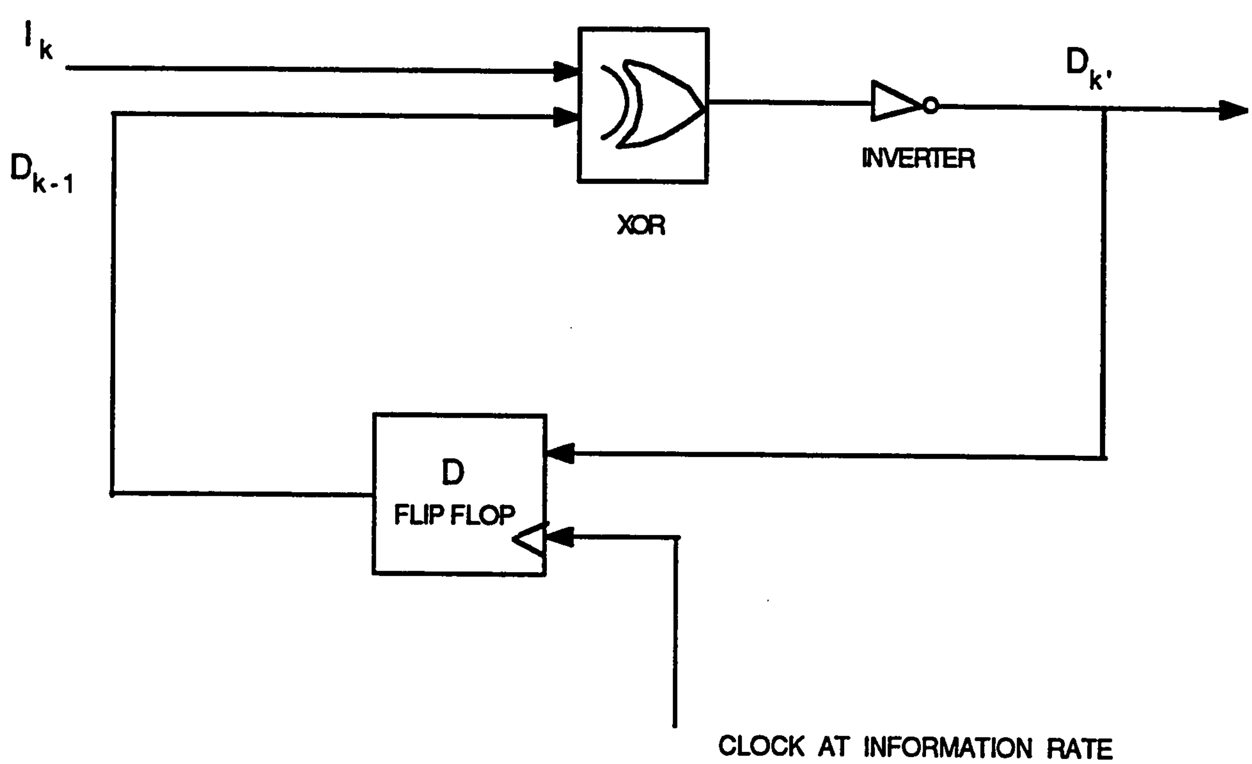
is then passed through a level circuit which provides the output D_k (+1 or -1) . This signal is then multiplied by a high speed code generator C_k to get the wide-spectrum output sequence :

$$A_k = C_k * D_k \qquad (3.3.2)$$



DPSK DSS TRANSMITTER BLOCK DIAGRAM

FIGURE # 3.1



LOGIC CIRCUIT

FIGURE # 3.2

The bit stream A_k is then used to modulate an intermediate frequency (IF) carrier using binary phase shift keying (BPSK) modulator. The modulated signal is then passed through a bandpass filter 1 (B.P.F 1) and the IF amplifier. The signal is then up-converted to radio frequency (RF) using a mixer and an RF oscillator . A Band Pass Filter 2 will then select the desired signal to be transmitted and the signal is then fed to a power amplifier and finally to the antenna.
The output transmitted signal is :

$$y(t) \simeq A_k * \cos((w_1 + w_2)t) \tag{3.3.3}$$

3.2.1 DIFFERENTIAL CODING

Differential coding of the input binary digits is used in the transmitter in order to enhance the system operation.
The followings are some advantages of the differential coding:

1. If phase hits occur on a BPSK system and the received data are 180^o out of phase the whole subsequent data will be error. In differential BPSK system no errors would occur since we are only concerned about the change

of phase between the received signal at time "t" and at " t-T ".

2. Simple non-coherent demodulation is used since there is no need for a reference carrier at the receiver.

Some disadvantages of the differential encoding are :

1. Error tends to propagate at least to adjacent bits.

2. Changing the transmission speed requires changing the delay element at the transmitter / receiver.

3. The signal power for DPSK should be more than that of BPSK for the same probability of error.

The probability of error for a differentially encoded binary phase shift keying is found to be [37]:

$$P_e = \frac{1}{2} * \exp^{(-Eb/No)} \qquad (3.3.4)$$

where,

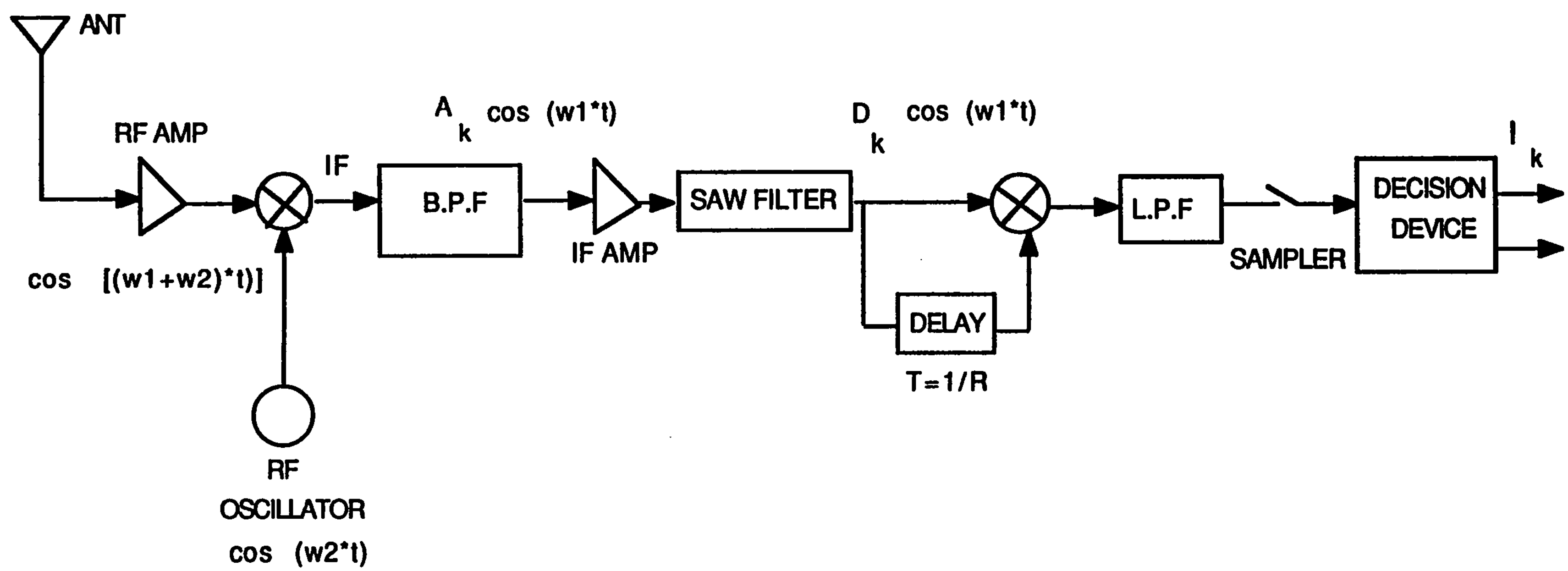
E_b = energy per bit

N_o = noise power spectral density in watt/Hz

Due to the non-coherent demodulation property and the 180^o phase hit error recovery, differential coding is recommended.

3.3 RECEIVER DESIGN

The demodulator circuit [36] is shown in Figure 3.3. The received signal is passed through an RF amplifier. The mixer and the RF oscillator are then used to down convert the RF signal into the IF stage. The IF signal is then passed through a SAW filter which correlates it's input with the pre-programmed code sequence generator (matching sequence). DPSK demodulation is then used to eliminate the need of synchronous carrier recovery at the receiver. In the DPSK demodulator loop a delay line T= 1/R , provides the one bit delay required in the demodulation process. The signal is then passed through a Low Pass Filter (L.P.F) which removes the harmonics of the IF carrier. Following the L.P.F is a sampler and a decision device used in order to recover our original transmitted information sequence B_k.



DPSK DSS RECEIVER BLOCK DIAGRAM
FIGURE # 3.3

3.4 COMPARISON OF VARIOUS CODE LENGTHS

The Federal Communication Commission (FCC) of U.S.A has authorized the use of spread spectrum communications in the industrial, scientific and medical bands. The frequency authorization includes 902-928 MHZ, 2400-2500 MHZ and 5725-5850 MHZ as long as the spread spectrum systems do not interfere with other licensees in these spectra [9].

Table 3.1 shows a comparison between various code lengths for the different Industrial, Scientific and Medical (ISM) bands authorized by the FCC.

DBPSK direct sequence spread spectrum system is assumed, and the following formulas were used to carry out the calculations [20] :

$$\text{Code rate} = (\text{ 3-db RF BW }) / 0.88 \qquad (3.5.1)$$

$$\text{Info rate} = \text{code rate} / \text{code length}$$

$$G_p = 10 * \text{LOG} (0.88 * \text{code length}) \qquad (3.5.2)$$

			CODE LENGTH = 31		CODE LENGTH = 63		CODE LENGTH = 127		CODE LENGTH = 255	
FCC ISM BANDS (MHZ)	3 - db RF BW (MHZ)	CODE RATE (MHZ)	INFO RATE (Mbit / sec)	Gp (db)	INFO RATE (Mbit / sec)	Gp (db)	INFO RATE (Kbit / sec)	Gp (db)	INFO RATE (Kbit / sec)	Gp (db)
902 - 928	26	29.545	0.953	14.358	0.469	17.438	232.64	20.483	115.86	23.53
2400 - 2500	100	113.636	3.665	14.358	1.804	17.438	894.77	20.483	445.63	23.53
5725 - 5850	125	142.045	4.5821	14.358	2.255	17.438	1118.5	20.483	557	23.53

TABLE 3.1

We notice that as the code length increases the information rate decreases and the process gain increases. Doubling the code length will give us a 3-db improvement in the process gain and will reduce our information rate to one half (1/2).

Error detection and correction codes could also enhance our system operation by reducing transmission errors. The effect of coding overhead introduced by various error correcting codes will increase the signal apparent data information rate and thus reducing the Jamming Margin. Assume the overhead bits are equal to the actual information bits, our process gain will be decreased by 3-db. Based on that, any error correcting technique employed in this example must have a coding gain of at least 3-db otherwise the system performance will be degraded.

In the next section, the proposed access procedure used for communication between the master station and the RTUs is described.

3.5 ACCESS PROCEDURE

Figure 3.4 shows a flowchart for the proposed access procedure used between the master station and the RTUs. The master station starts the polling cycle by looking at its polling table which contains the addresses of all RTUs to be polled. After the master requests the I'th RTU to send its packet, it waits an " alpha " second before it polls the second RTU. If the master receives a correct packet it sets the n'th bit of the control frame, for that particular RTU , to one (1) in order to clear that packet from the RTU buffer at its next polling cycle. On the other hand, if the master does not receive a correct packet, it sets the n'th bit of the control frame to zero so that the master receives the same packet on its next polling cycle. Furthermore, the master will remove an RTU from its polling table when a particular RTU does not respond or keeps sending incorrect packets for the master station . See Figure 3.4 .

The following conditions are assumed in our access procedure:

1. Each RTU has a different address and the master station will poll the RTUs according to its polling table .

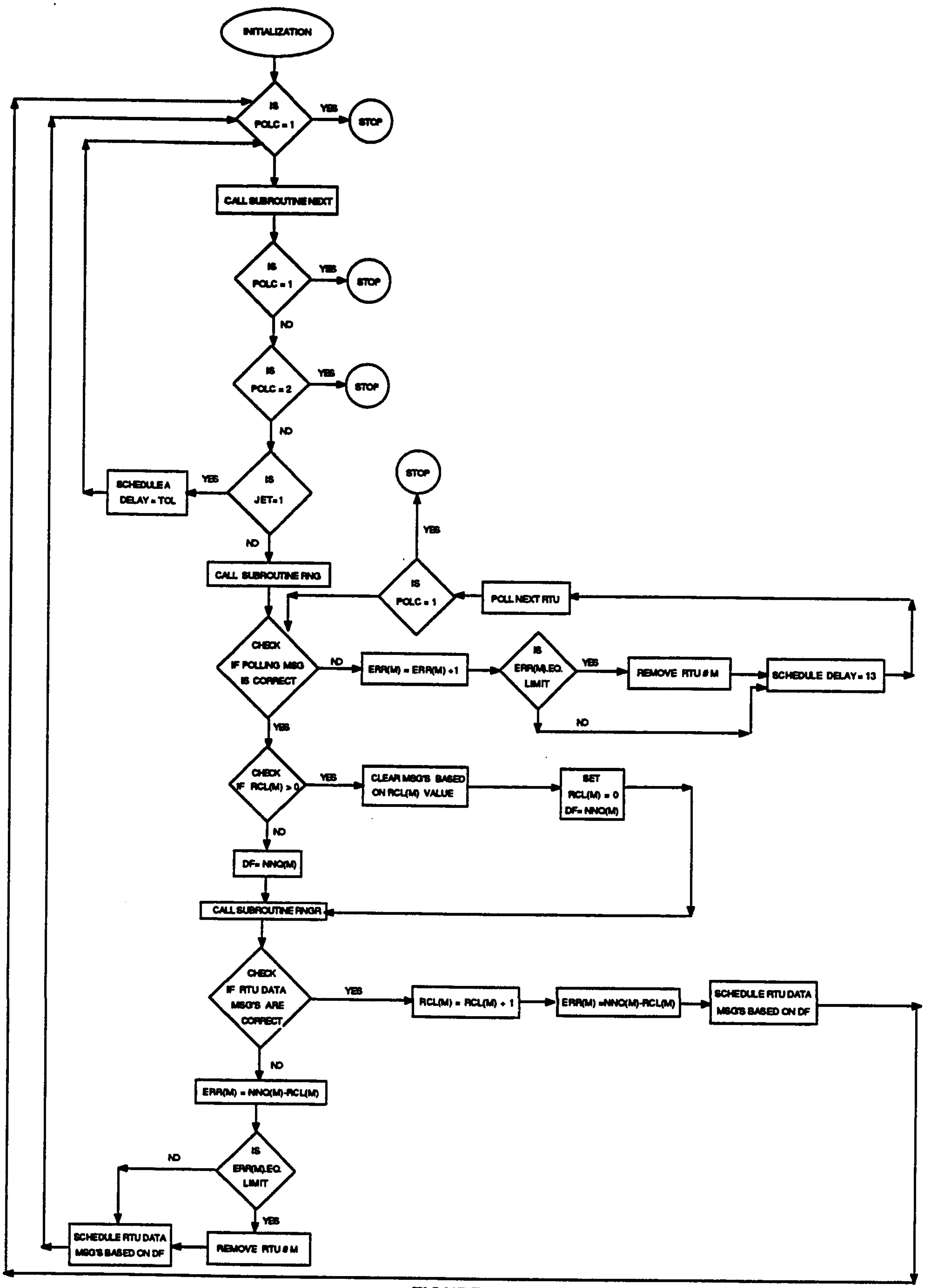

INITIALIZATION
IS POLC = 1
YES
STOP
CALL SUBROUTINE NEXT
IS POLC = 1
YES
STOP
NO
IS POLC = 2
YES
STOP
NO
IS JET=1
YES
SCHEDULE A DELAY = TCL
NO
CALL SUBROUTINE RNG
STOP
YES
IS POLC = 1
POLL NEXT RTU
CHECK IF POLLING MSG IS CORRECT
NO
ERR(M) = ERR(M) +1
IS ERR(M).EQ. LIMIT
YES
REMOVE RTU # M
SCHEDULE DELAY = 13
NO
YES
CHECK IF RCL(M) > 0
YES
CLEAR MSG'S BASED ON RCL(M) VALUE
SET RCL(M) = 0 DF= NNO(M)
NO
DF= NNO(M)
CALL SUBROUTINE RNGR
CHECK IF RTU DATA MSG'S ARE CORRECT
YES
RCL(M) = RCL(M) + 1
ERR(M) =NNO(M)-RCL(M)
SCHEDULE RTU DATA MSG'S BASED ON DF
NO
ERR(M) = NNO(M)-RCL(M)
IS ERR(M).EQ. LIMIT
NO
YES
SCHEDULE RTU DATA MSG'S BASED ON DF
REMOVE RTU # M

FIGURE # 3.4

2. " Alpha " seconds is the time which the master station should wait before it can poll another RTU.

$$\text{Alpha (sec)} = Y + T_t + T_m \qquad (3.1.1)$$

Y = Round-trip propagation delay.

T_t = Transmission time for the polling and RTU data messages

T_m = Processing time at the RTU and the master station.

3. All packets generated within a specific RTU are of equal importance .

4. RCL(I) is an array element used by the master station and sent by its polling message, inside the control frame, to acknowledge the previously received packets of the I'th RTU.

   - If RCL(I) = L , where L is an integer greater than or equal to one, then the master station will clear L packets of the I'th RTU buffer and request the RTU to send the remaining packets.

- If RCL(I)=0 , the master station will not clear any packects from the I'th RTU and it will request the RTU to send all the packets available at its buffer.

5. Each RTU will generate one packet of data during one scanning cycle.

6. The processing time for the host computer and each of the RTUs is 100 micro-sec.

7. The polling message (Master to RTU) is either ten bytes or thirty two bytes long and the data message (RTU to Master) is sixty four bytes long.

8. The distance between each RTU and the central computer is identical. Therefore, the total propagation delay is :

$$Y' = Y * n \qquad (3.1.2)$$

where n is the number of active RTUs.

9. There are always data messages in each RTU buffer.

10. Assume the polling rate is 10 cycles/sec. Therefore, each cycle is equal to 100 msec.

11. RTU messages should be acknowledged in sequence. That is, if two packets are to be transmitted from one RTU and the second packet was received correctly, whereas the first packet was in error, then we retransmit both packets again.

12. The master station should have the ability to transmit control commands to the various RTUs.

In the next section, the frame format used in our work is described.

3.6 FRAME FORMAT

This section illustrates the proposed frame format to be used in our work. It is based on the IEEE standard 802.4 Token - passing on a bus [38].

Preamble	SD	FC	DA	SA	INFO	FCS	ED

Preamble = is one or more bytes long

SD = start delimiter is one byte long.

FC = control frame is one byte long.

DA = destination address is two bytes long.

SA = source address is two bytes long.

INFO = information frame is variable limited by the maximum allowable frame size.

FCS = frame check sequence is two bytes long.

ED = end delimiter is one byte long.

As noticed from the above frame structure, the Frame Check Sequence (FCS) is two bytes long whereas in the IEEE 802.4 it is four bytes long. The main reason for decreasing the length of the FCS, from four bytes to two, is to decrease the number of overheads in the frame structure.

Explanation of each of the above fields is described as follows :

PREAMBLE

Preamble should be at least 2 micro-sec regardless of the data rate. This delay time is used to allow a particular RTU to process its previously received packets.

START DELIMITER

The frame structure requires a start delimiter, which begins the frame.

FRAME CONTROL (FC)

Is an 8-bit sequence C1 C2 C3 C4 C5 C6 C7 C8

<u>C1</u>	<u>C2</u>	
1	1	Special Purpose Frame (S.P.F)
0	X	Information Frame (I.F), where x is either 0 or 1.
1	0	Supervisory Frame (S.F)

The following commands should be included in the frame control field [25] :

" Reset " a particular RTU

" Test " analog loopback , digital loopback and self test

" XID " exchange identification

" DISC " disconnect RTU

" SIM " set initialization mode

" NRM " normal response mode (unbalanced config.) : The master station may initiate data transfer to a secondary, but the secondary may only transmit data in response to a poll from the master.

In addition to the above mentioned commands, we should also include two other commands required for the new proposed access procedure.

" ACK " Master station acknowledges the previous RTU packet/s and clears the top packet/s at the RTU buffer.

" NACK " Master does not acknowledge the reception of the previously transmitted RTU packet/s.

Destination Address

The destination address is a 16-bit sequence I1 through I16

<u>I1</u>

0 Individual address : Is used by the master station to address one station .

1 Group Address : Is used by the master station to address group of stations .

If all entries of the address field are ones, then such address is known as the broadcast address. This address is used by the master station to synchronize the whole network. Furthermore, the address field corresponding to all zero's is used for testing.

The 15 bits I2 through I16 are available for addressing purposes, out of fifteen bits we can get 2^{15} different combinations .

Number of available destination addresses = $2^{15} - 2 = 32766$

Source Address

The source address contains the identification address of the source station.

Frame Check Sequence (FCS)

Is a 16-bit Cyclic Redundancy Check (CRC) sequence. The generator polynomial g(x) is the standard polynomial used in High Data Link Control (HDLC) protocol [39].

$$g(x) = x^{16} + x^{12} + x^{5} + 1 \tag{3.2.1}$$

This polynomial catches all single and double errors, all errors with an odd number bits and all burst errors of length 16 or less.

End Delimiter

The end delimiter is used to indicate the end of the frame .

SUMMARY

In this chapter, the transmitter and receiver circuits are designed using "off the shelf" components. Then a new access procedure for communication between the master station and the RTUs is introduced. Finally, the IEEE 802.4 frame format is recommended.

In the next chapter, the overall system performance is evaluated and simulation results are obtained using Simulation Language Alternate Modelling II (SLAMII) software package (Appendix A).

CHAPTER 4

PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM USING NETWORK SIMULATION

In this chapter, the overall network performance of the proposed RF spread spectrum system is evaluated under various probabilities of errors and for different values of n (number of RTUs). Furthermore, the effect of the transmission rate on the system performance is also considered. In the next sections, the simulation objectives, procedure and results are presented.

4.1 SIMULATION OBJECTIVES

The objectives of the network simulation program can be summarized as follows :

1. Study the effect of Probability of error P_e against System Idle Time for various values of n (number of RTUs).

2. Calculate the Average Message Delay Time (AMDT) in the

RTU buffer.

3. Investigate the usefulness of increasing or decreasing the data rate.

4 Repeat system analysis when control commands are activated by the host computer.

In general, the main objective of this thesis is to assist network users in designing their own network. This can be done by providing the users with various design curves which they can utilize depending on their network environment and parameters.

4.2 SIMULATION PROCEDURE

Figure 3.4 illustrates the flowchart used in writing the simulation program with the source code given in Appendix B & C. The software package used, in writing the program, is Simulation Language Alternate Modelling II (SLAM II) [40].

Below are some of the steps followed in order to simulate the system performance :

1. Choose a certain probability of error.

2. Choose a certain value of n = number of RTUs.

3. Record the value of the System Idle Time (SIT) at each polling cycle.

4. Find the Minimum System Idle Time (MSIT), average message length and average delay time for all cycles.

5. Change the probability of error and repeat step 2 to 4.

4.3 SIMULATION RESULTS

During the network simulation, the average buffer size, average delay time and system idle time are obtained at several values of "n" (number of RTUs) with probability of packet error as a parameter in the simulation runs.
Two different situations are considered for simulation.
These are :

1) Data Acquisition
2) Data Acquisition & Control

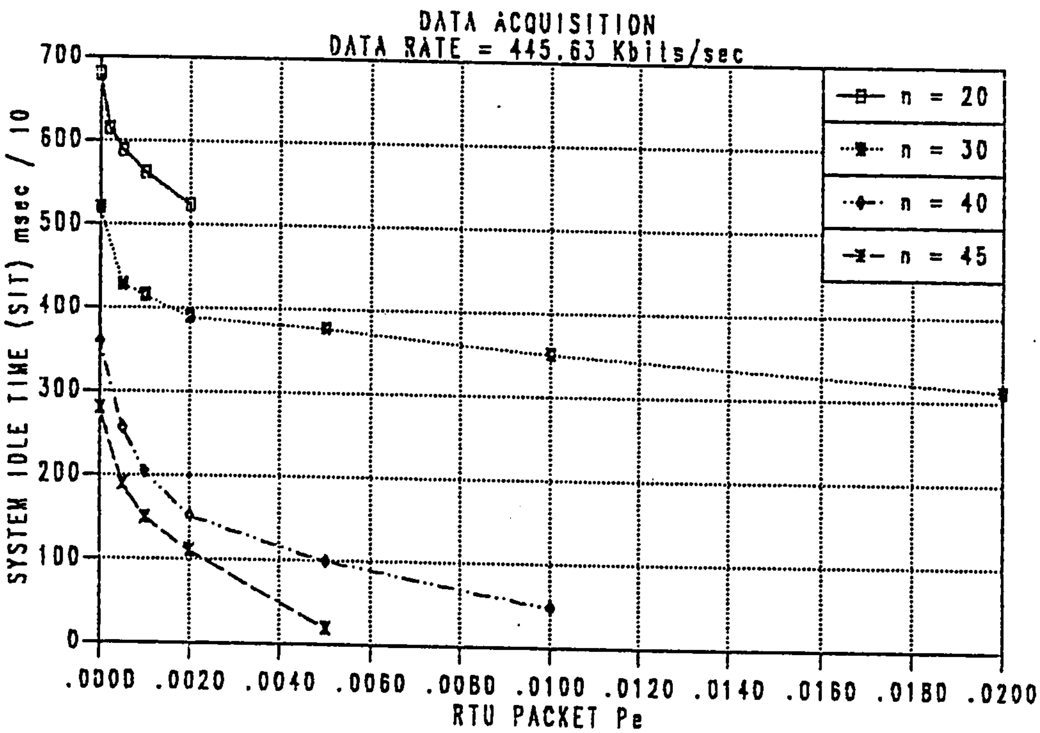
4.3.1 Data Acquisition

In this case, the RTUs are only responsible for collecting data information from the field and sending it to the master station. The master station polling message is ten bytes long and does not contain an information field.

Figure 4.2 through 4.5 are the simulation results for data rates at 445.63 & 894.77 Kbits/sec.

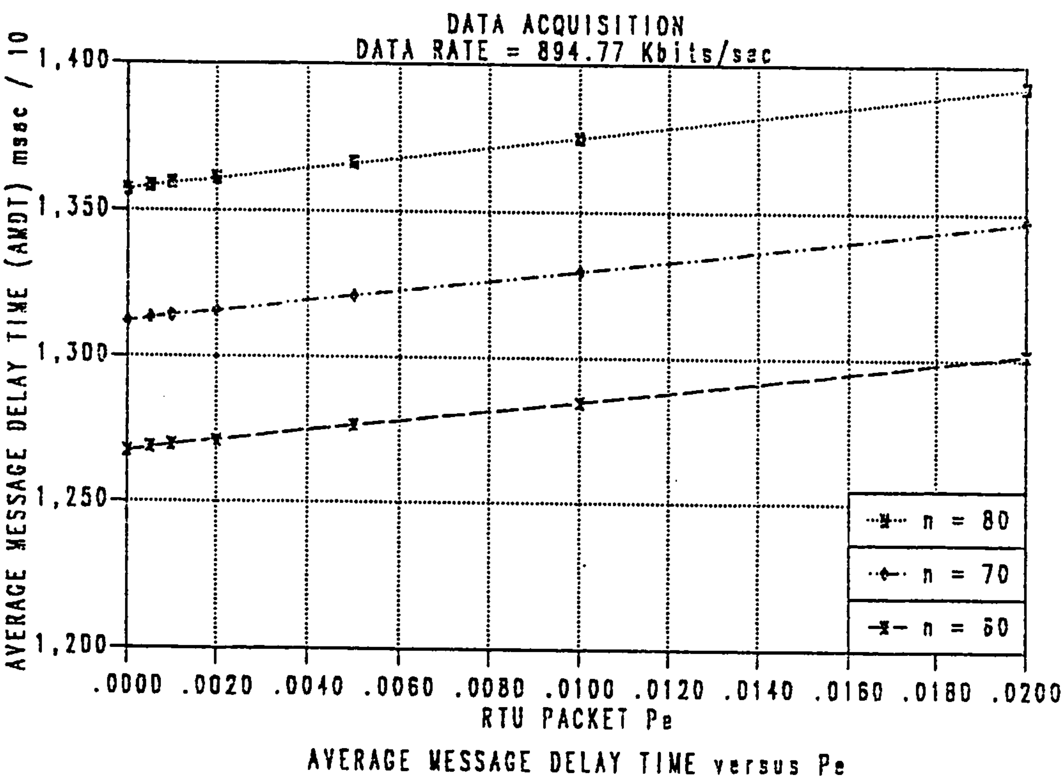
In Figure 4.1 the System Idle Time (SIT) is plotted versus RTU packet error (P_e). As noticed from the graph, when P_e increases, the system idle time decreases. This result is expected because when RTU data packets are received incorrectly by the master station, the host computer will then request a retransmission of the wrong data packets in the next polling cycle. Therefore, the system idle time is decreased. Furthermore, as the number of RTUs used in the network increases, the system idle time decreases. Also, when the number of RTUs is equal to 20 , we notice that we cannot operate at high error rates. This is due to the limitation on the RTU buffer size which causes some RTUs to be removed from the polling cycle. Moreover, when n=40 we cannot also operate at high error rates since the SIT will be negative, which indicates that polling period exceeds one cycle .

Figure 4.2 shows a graph of the average RTU message delay time against P_e for different number of RTUs. We notice that, as the P_e increases, more messages have to wait in the RTU buffer for retransmission, thus causing the average message delay time to increase.



SYSTEM IDLE TIME versus Pe

FIGURE 4.1



DATA ACQUISITION
DATA RATE = 894.77 Kbits/sec
AVERAGE MESSAGE DELAY TIME (AMDT) msec / 10
1,400
1,350
1,300
1,250
1,200
.0000 .0020 .0040 .0060 .0080 .0100 .0120 .0140 .0160 .0180 .0200
RTU PACKET Pe
n = 80
n = 70
n = 60
AVERAGE MESSAGE DELAY TIME versus Pe

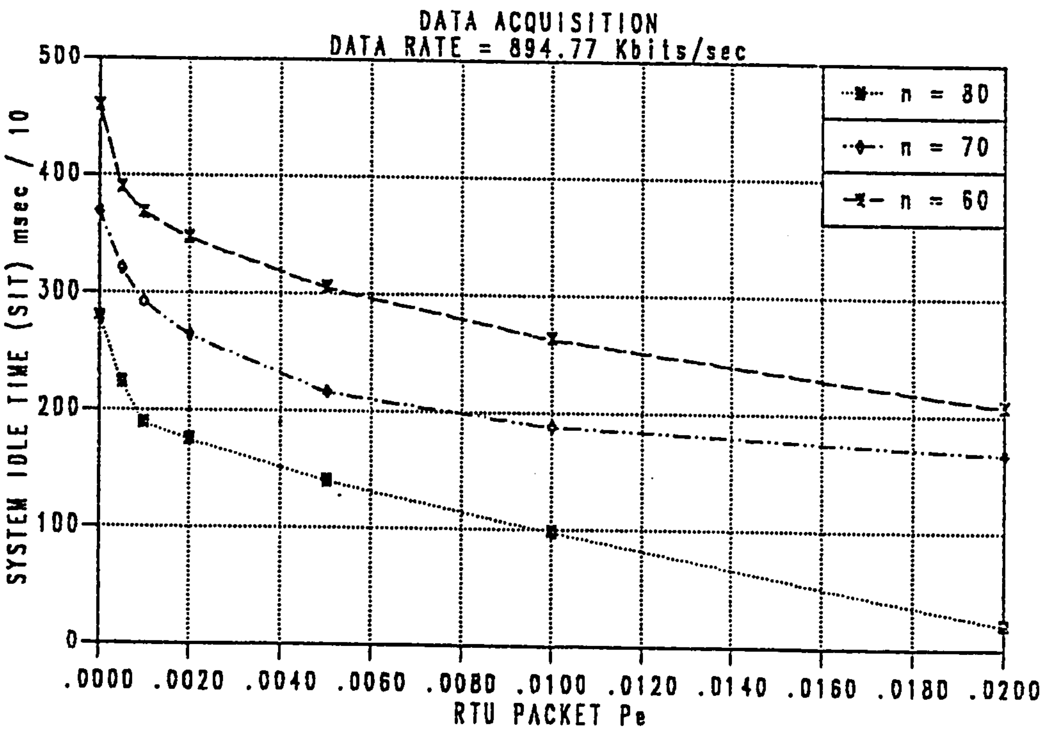
FIGURE 4.2

In Figure 4.3 and 4.4, the data rate is increased to 894.77 Kbits/sec and the following results are observed :

1. More RTUs can be supported for the same SIT & P_e. This is due to the higher transmission speed which causes more RTU data packets to be transmitted with the same delay time.

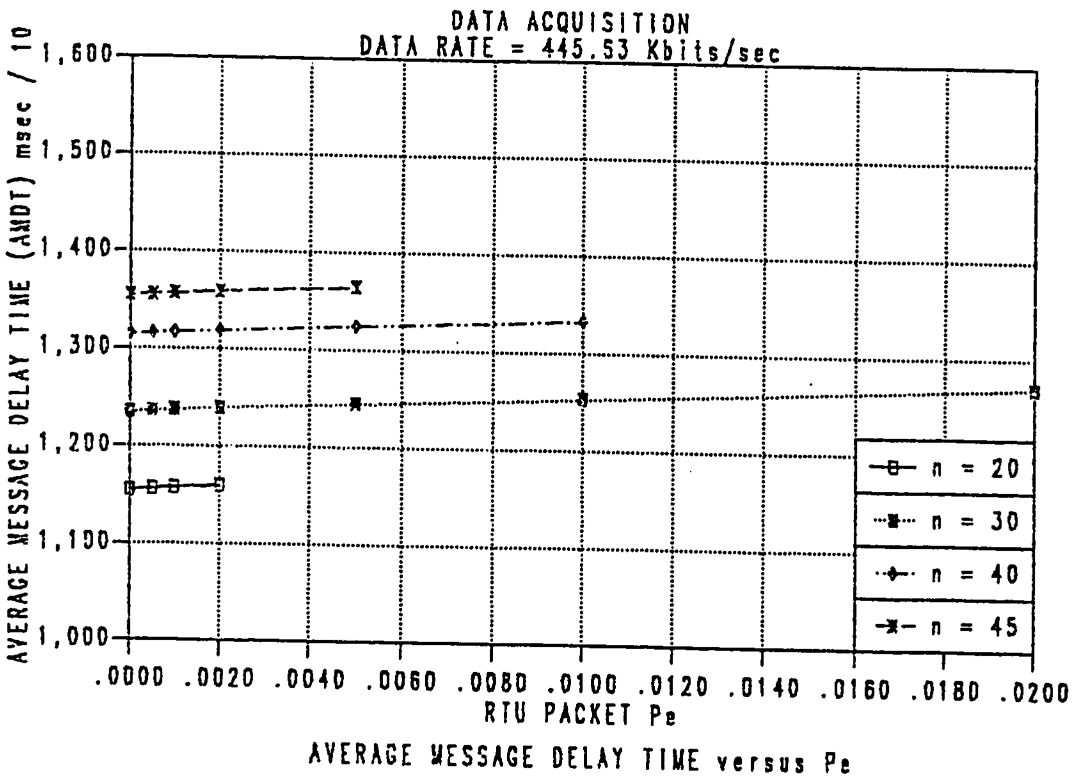
2. For the same P_e, about twice the number of RTUs are supported with nearly the same AMDT. This is because the master station polls RTUs data packets at a faster speed and hence less delay time.

Moreover, data rates higher than 894.77 Kbits/sec (see Table 3.1) can also be considered in the network simulation. This can be achieved by increasing the code rate of the code sequence generators in the transmitter and receiver circuitry of the proposed RF spread spectrum system.



SYSTEM IDLE TIME versus Pe

FIGURE 4.3



DATA ACQUISITION
DATA RATE = 445.53 Kbits/sec
AVERAGE MESSAGE DELAY TIME (AMDT) msec / 10
1,600
1,500
1,400
1,300
1,200
1,100
1,000
.0000 .0020 .0040 .0060 .0080 .0100 .0120 .0140 .0160 .0180 .0200
RTU PACKET Pe
n = 20
n = 30
n = 40
n = 45
AVERAGE MESSAGE DELAY TIME versus Pe

FIGURE 4.4

4.3.2 Data Acquisition & Control

In addition to data acquisition, the master station can send control commands to the various RTUs while polling them. The polling message is increased to thirty two bytes where twenty two bytes out of them are allocated to the information field.

The following additional parameters are included in the simulation :

1. INRC : Is the total number of RTUs which accept control commands.

2. RCOM(I) : Is an array which contains the address of each RTU allowed to accept control commands.

In Figure 4.5 through 4.8 we assume that twenty and fifty percent of the total RTUs can accept control commands, in addition to data acquisition. The data rate is assumed to be 445.63 Kbits/sec

We notice that the system idle time is now decreased significantly as compared to figures 4.1. This is due to the

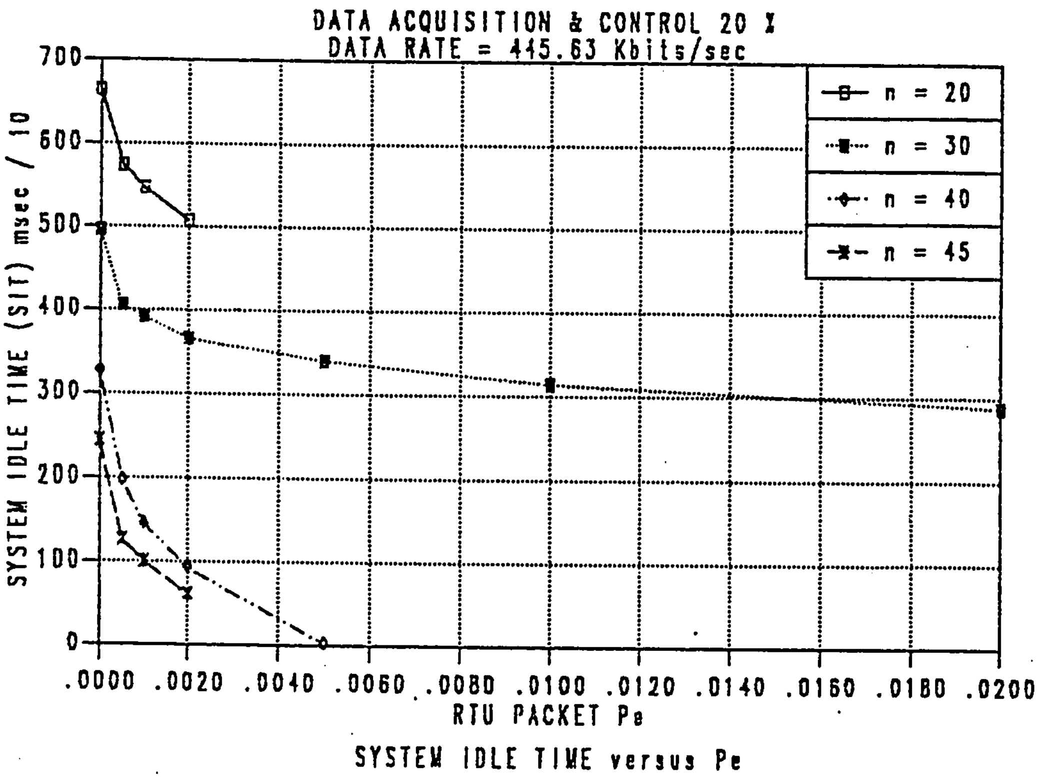

DATA ACQUISITION & CONTROL 20 %
DATA RATE = 445.63 Kbits/sec
SYSTEM IDLE TIME (SIT) msec / 10
700
600
500
400
300
200
100
0
n = 20
n = 30
n = 40
n = 45
.0000 .0020 .0040 .0060 .0080 .0100 .0120 .0140 .0160 .0180 .0200
RTU PACKET Pe
SYSTEM IDLE TIME versus Pe

FIGURE 4.5

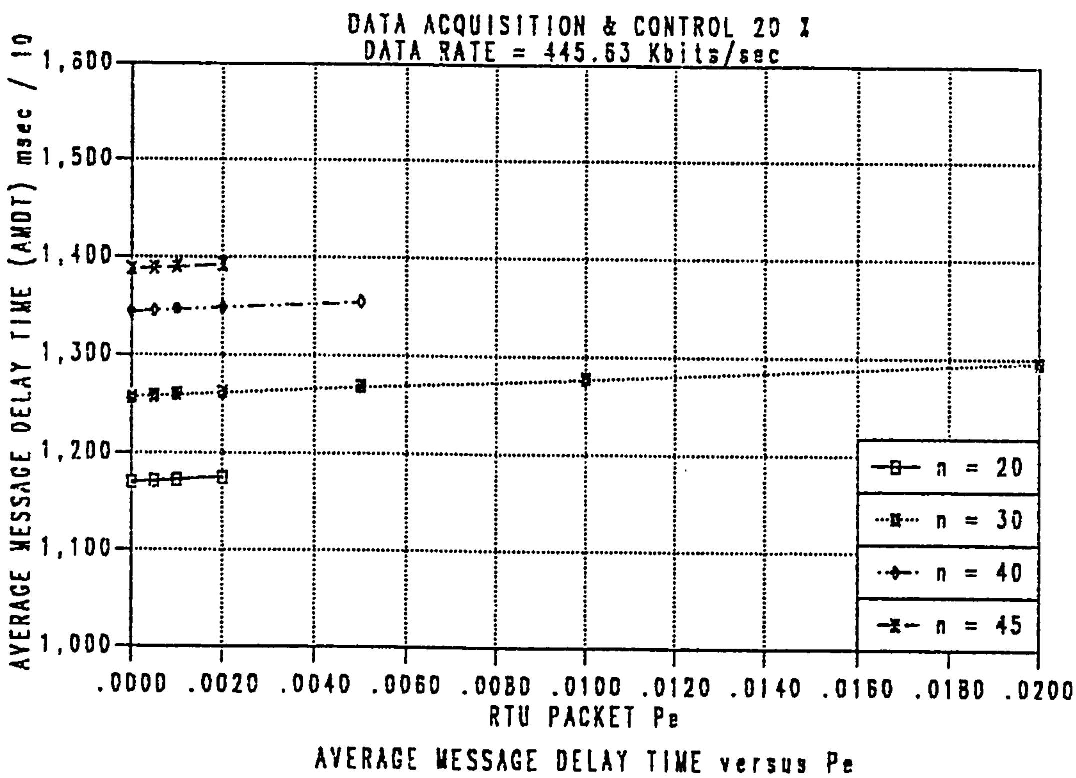
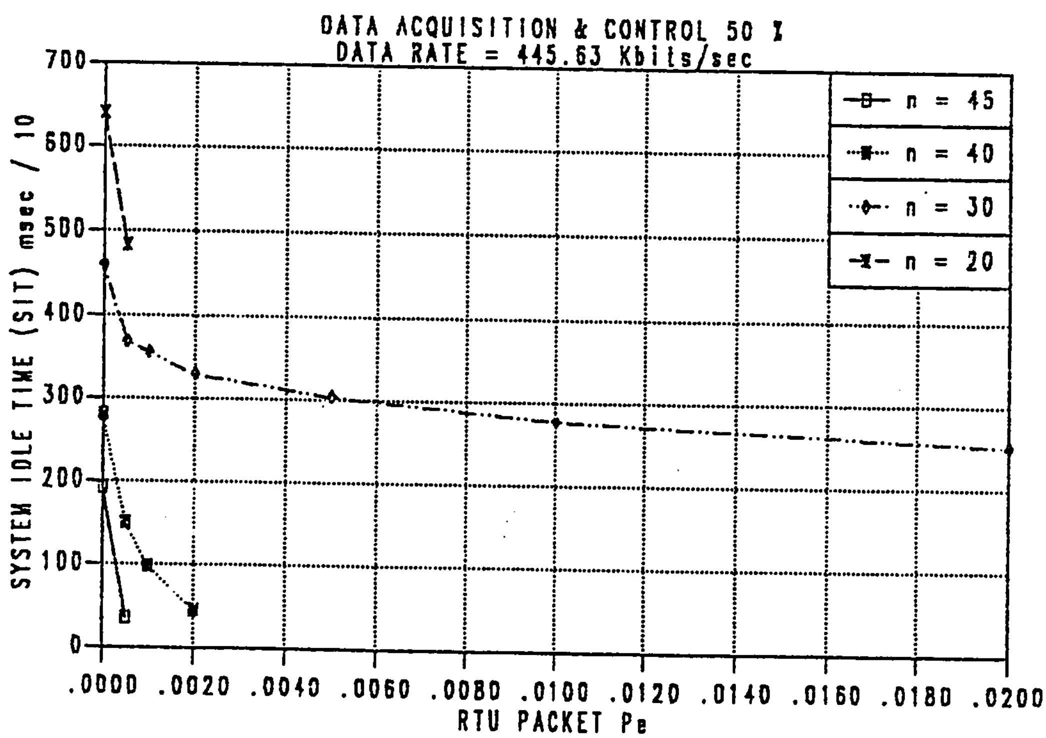
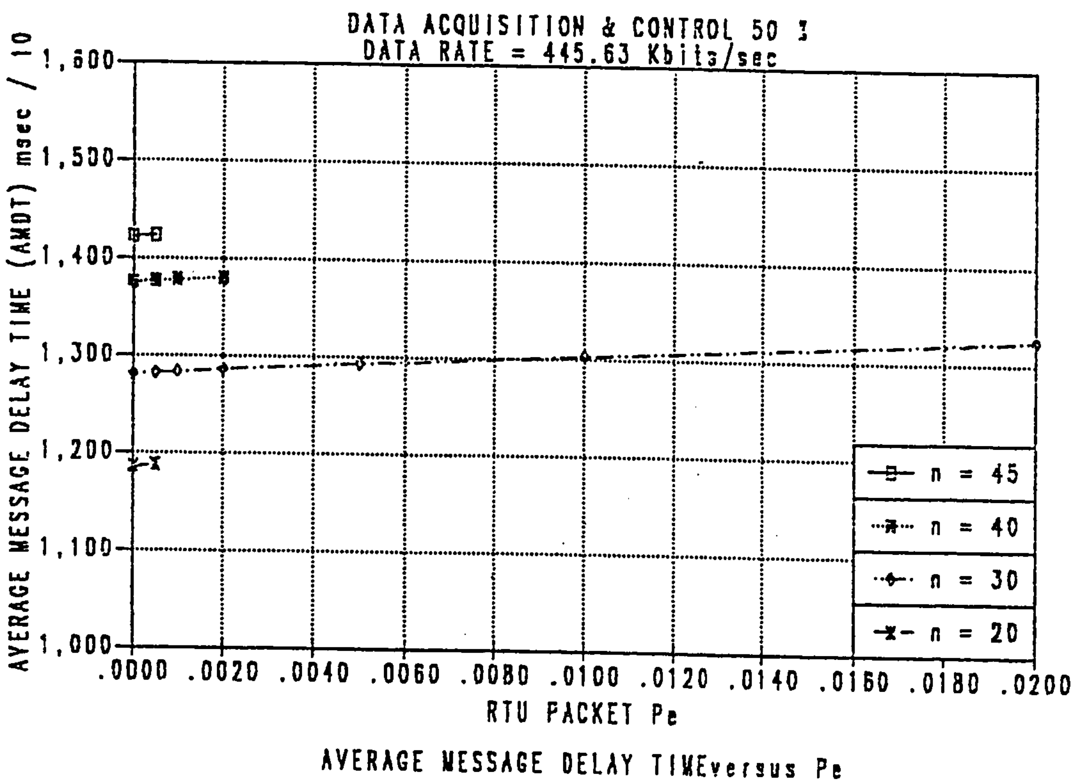

DATA ACQUISITION & CONTROL 20 %
DATA RATE = 445.63 Kbits/sec
AVERAGE MESSAGE DELAY TIME (AMDT) msec / 10
1,600
1,500
1,400
1,300
1,200
1,100
1,000
.0000
.0020
.0040
.0060
.0080
.0100
.0120
.0140
.0160
.0180
.0200
RTU PACKET Pe
n = 20
n = 30
n = 40
n = 45
AVERAGE MESSAGE DELAY TIME versus Pe

FIGURE 4.6



SYSTEM IDLE TIME versus Pe

FIGURE 4.7



DATA ACQUISITION & CONTROL 50 %
DATA RATE = 445.63 Kbits/sec
AVERAGE MESSAGE DELAY TIME (AMDT) msec / 10
1,600
1,500
1,400
1,300
1,200
1,100
1,000
.0000 .0020 .0040 .0060 .0080 .0100 .0120 .0140 .0160 .0180 .0200
RTU PACKET Pe
n = 45
n = 40
n = 30
n = 20
AVERAGE MESSAGE DELAY TIMEversus Pe

FIGURE 4.8

increase in the polling message length from ten bytes, in case of data acquisition, to thirty two bytes when control commands are considered. Furthermore, the average message delay time is increased as compared to Figure 4.2.

SUMMARY

In this chapter, the system performance is evaluated for two different cases, data acquisition, and data acquisition and control. In both cases, we have determined the system idle time and the average message delay time under various bit-error rates. In general, we notice that, as the probability of error increases the system idle time decreases and the average RTU message delay time increases. Furthermore, it is found that sometimes the system idle time is negative under certain values of n (number of RTUs) and P_e. This situation indicates that polling period exceeds the one cycle limit (100 msec).

CHAPTER 5

CONCLUSIONS AND SUGGESTIONS FOR FURTHER STUDIES

In this chapter, the main thesis results are summarized, followed by suggestions for possible future work extension.

5.1 CONCLUSIONS

A new application of radio frequency spread spectrum system is proposed for Distributed Control Systems. The new method eliminates the need of excessive use of cable wiring and provides an increased flexibility and data security. It uses a new polling access protocol for communication between field RTUs and master station. The system performance is evaluated for different data rates and various bit-error probabilities. The followings are the main thesis results :

1. A system with a small number of RTUs should operate at low probability of error due to the buffer size limitation.

2. As the number of RTUs increases, the system idle time decreases significantly.

3. The average message delay time is found to be a maximum of about 1.42 * cycle time.

4. On the average, the RTUs do not require a large buffer size.

5. As the data rate increases, more RTUs can be supported.

5.2 SUGGESTIONS FOR FURTHER STUDIES

The work presented in this thesis may be extended for further research in the following aspects :

1. Increase the system capacity (number of RTUs) by adding another group of RTUs (cell) which have similar transmitters and receivers but different code sequence generator from the previous cell. Simulate the system for multi-cell situation.

2. Vary the maximum buffer size capacity of the RTUs and repeat the whole simulation.

3. Change the RTU data message length and simulate the network.

4. Study the effect of varying the code sequence generator on the system performance.

5. Study a full-duplex system where the master station and the RTUs can transmit and receive at the same time.

6. Use a DQPSK or DQAM instead of DBPSK.

APPENDIX A

SIMULATION LANGUAGE ALTERNATE MODELLING II (SLAMII)

SLAMII simulation software package is applied to model our network. In this package, there are three approaches which the network modeler can select :

1. Network Orientation method.
2. Discrete Event Modelling.
3. Continuous Event Modelling.

In this work, we have selected the Discrete Event Modeling since we are dealing with discrete events rather than continuous time events. Moreover, the discrete event approach is more flexible than the network orientation method due to its ability to interface easily with FORTRAN.

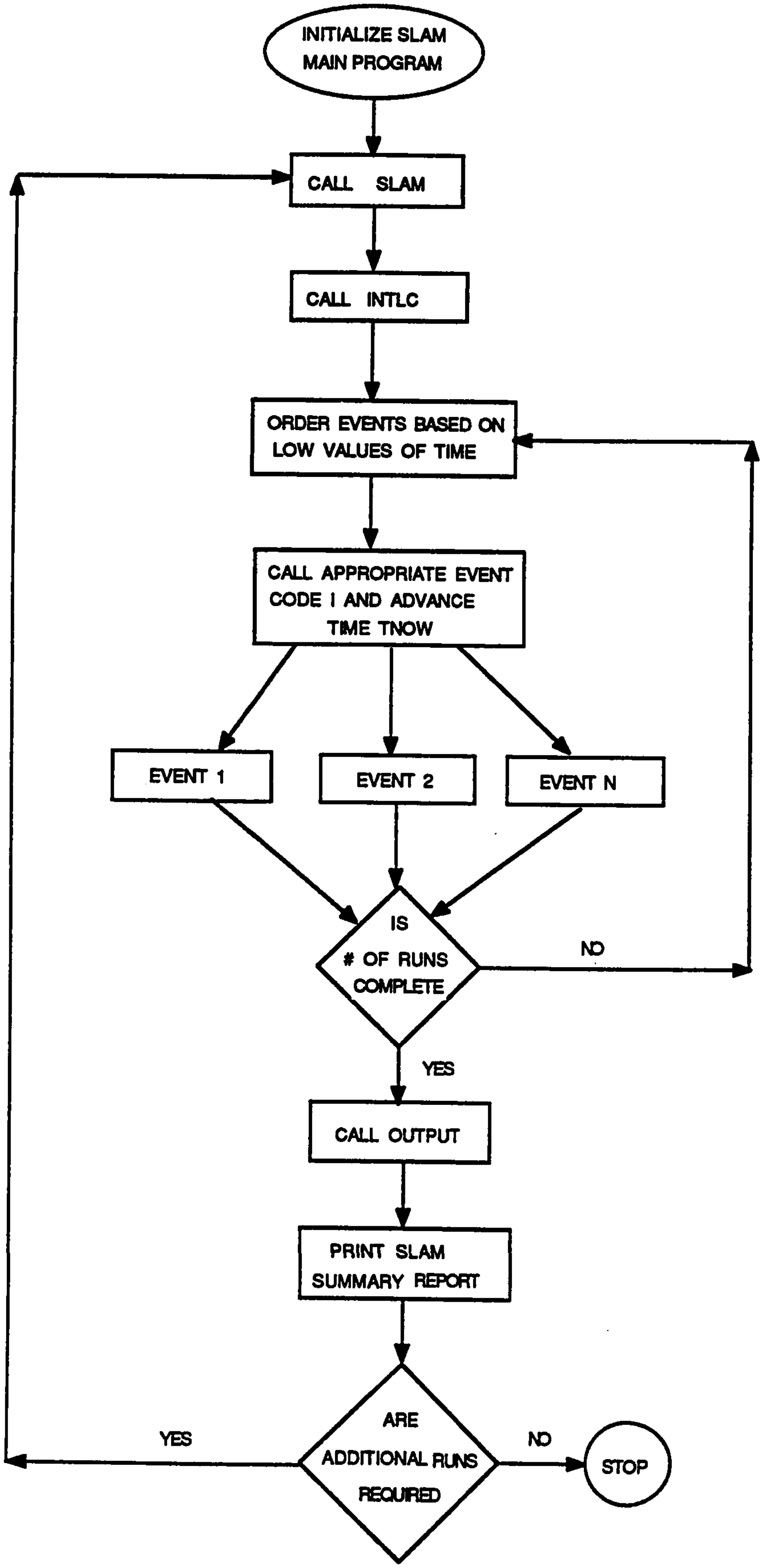
In order to model a discrete event network, the user should start by writing the main program which calls SLAM subroutine. This subroutine acts as the executive control for the whole network. It is responsible for various activities such as event scheduling (e.g.: subroutine SCHDL) and statistics collection (e.g.: subroutine COLCT). Therefore, the network modeler is relieved from the task of sequencing the various events according to their corresponding time order (TNOW).

There are two additional user written subroutines which are usually included in most SLAM discrete event models. These are subroutines INTLC and OTPUT.

Subroutine INTLC is called by SLAM before each simulation run and is used to set the initial conditions and schedule initial events.

Subroutine OTPUT is called at the end of each simulation run and is used to output the results.

Figure A shows a simplified flowchart for simulating discrete event models.



SLAMII DISCRETE EVENT MODELLING

FIGURE # A

APPENDIX B

SLAM II SIMULATION PROGRAM FOR
DATA ACQUISITION

M.S. THESIS

RADIO FREQUENCY SPREAD SPECTRUM
FOR
DISTRIBUTED CONTROL SYSTEMS

Done By : **G. B. AL-DANDAN**

THIS PROGRAM IS WRITTEN TO POLL N DIFFERENT RTUs. THE MASTER STATION INTERROGATES THE VARIOUS RTUs IN A SEQUENTIAL MANNER. THE SCANNING CYCLE IS 1000 TIME UNITS WHICH IS EQUIVALENT TO 100 msec. THE TRANSMISSION DELAY DEPENDS ON THE DATA RATE USED AND THE NUMBER OF MESSAGES IN THE RTU BUFFER. THE MINIMUM SYSTEM IDLE TIME AND THE AVERAGE BUFFER DELAY IS FOUND DURING THE SIMULATION.

VARIABLES DEFINITION

ERR(M) : IS THE NUMBER OF DATA MESSAGES THAT ARE LOCATED IN THE M'TH RTU AND HAVE NOT BEEN TRANSMITTED TO THE MASTER STATION DUE TO TRANSMISSION ERROR CAUSED BY EITHER WRONG POLLING MSG OR WRONG RECEPTION OF RTU DATA MSG.

RCL(M) : IT CONTAINS THE NUMBER OF MESSAGES THAT SHOULD BE REMOVED FROM THE M'TH RTU WHEN THE CORRECT POLLING MSG REACHES THAT PARTICULAR RTU.

NNQ(M) : IS THE TOTAL NUMBER OF MSG'S AVAILABLE AT THE M'TH RTU AND AT THE CURRENT TIME (SLAMII VARIABLE).

DST(JS) : IT CONTAINS THE NUMBER OF RTUS WHICH HAVE BEEN REMOVED FROM THE POLLING CYCLE.

TOL : IS THE TIME WHICH THE MASTER STATION SHOULD WAIT AT THE END OF EACH POLLING CYCLE (TOLERANCE).

RTN : THE TOTAL NUMBER OF RTUS IN THE NETWORK.

JET	:	1 INDICATES MASTER STATION SHOULD STOP POLLING THE RTUS AND WAIT A TIME SPECIFIED BY " TOL " VALUE.
POLC	:	1 INDICATES THAT ALL RTUS HAVE BEEN REMOVED FROM THE POLLING CYCLE. 2 INDICATES THAT THE TOLERANCE IS NEGATIVE.
LIMIT	:	IS THE MAXIMUM NUMBER OF ERROR MSG'S ALLOWED BEFORE A PARTICULAR RTU IS REMOVED FROM POLLING CYCLE.
ELMTP	:	A CERTAIN VALUE IF EXCEEDED THE POLLING MSG WILL BE IN ERROR.
ELMTR	:	A CERTAIN VALUE IF EXCEEDED THE RTU DATA MSG WILL BE IN ERROR.
PER	:	PACKET PROBABILITY OF ERROR FOR THE RTU DATA MSG.
SP	:	IS THE TRANSMISSION SPEED OR THE DATA RATE USED.
MAX	:	IS THE MAXIMUM NUMBER OF RTUS USED + 2

```
$STORAGE:2
$NOTSTRICT
$PAGESIZE:60
$NOTLARGE
$NOFLOATCALLS
  PROGRAM MAIN
  DIMENSION NSET(20000)
  COMMON/SCOM1/  ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
 1,MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
 1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  COMMON QSET(20000)
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  EQUIVALENCE(NSET(1),QSET(1))
  NNSET=20000
  NCRDR=5
  NPRNT=6
  NTAPE=7
  CALL SLAM
  STOP ' '
  END

 C ****************************************
 C              THE EVEVT SUBROUTINE
 C ****************************************
  SUBROUTINE EVENT(I)
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
      GO TO (1,2,3,4,5,6,7,8,9) ,I
 1    CALL POLL
      RETURN
 2    CALL RELAX
      RETURN
 3    CALL ENDSV
      RETURN
 4    CALL INPUT
```

```
      RETURN
5     CALL TEST
      RETURN
6     CALL HELP
      RETURN
7     CALL TRIAL
      RETURN
8     CALL REMOVAL
      RETURN
9     CALL SAVE
      RETURN
      END
C ****************************************
C       THE INITIALIZATION SUBROUTINE
C ****************************************
 SUBROUTINE INTLC
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
1,ELMTP,ELMTR
 DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
 PER=
 ELMTP=
 ELMTR=
 SP=
 LIMIT=4
 MAX=42
 MIN=3
 M=MIN-1
 JS=1
 POLC=0
 TOL=0
 JET=0.0
 NC=1
 INT=1
C
C INITIAL REGISTER VALUES OF THE RANDOM NUMBER GENERATOR
```

```
C (IRAND)
C
 ISR(1)=0
 ISR(2)=1
 ISR(3)=0
 ISR(4)=0
 ISR(5)=0
 ISR(6)=0
 ISR(7)=0
 ISR(8)=0
 ISR(9)=0
 ISR(10)=0
 ISR(11)=0
 ISR(12)=0
 ISR(13)=0
 ISR(14)=1
 ISR(15)=0
 ISR(16)=0
 ISR(17)=0
 ISR(18)=0
 ISR(19)=0
 ISR(20)=0
 ISR(21)=0
 ISR(22)=0
C
C INITIAL REGISTER VALUES OF THE RANDOM NUMBER GENERATOR
C (IRANDR).
C
 IS(1)=0
 IS(2)=0
 IS(3)=0
 IS(4)=0
 IS(5)=0
 IS(6)=0
 IS(7)=0
 IS(8)=0
 IS(9)=0
 IS(10)=0
 IS(11)=0
```

```
      IS(12)=0
      IS(13)=0
      IS(14)=0
      IS(15)=0
      IS(16)=0
      IS(17)=1
      IS(18)=0
      IS(19)=0
      IS(20)=0
      IS(21)=0
      IS(22)=0
      IS(23)=0
C
         DO 10 J=1,150
                ERR(J)=0
                RCL(J)=0
                DST(J)=0
 10   CONTINUE
C
      TL=LIMIT
      CALL COLCT(TL,4)
      CALL COLCT(PER,5)
      RTN=(MAX-MIN)+1
      CALL COLCT(RTN,6)
      CALL COLCT(SP,7)
C
C      PUT ONE MSG IN EVERY RTU FILE AT T=0.0
C
       DO 30 I=1,1
          DO 20 J=MIN,MAX
             CALL FILEM(J,I)
 20       CONTINUE
 30    CONTINUE
C   SCHEDULE EVENT 1 TO START AT TIME=0.0 AND INPUT DATA
C   MESSAGES TO THE RTU'S USING EVENT 4 EVERY 999 TIME UNITS
      CALL SCHDL(1,0.0,ATRIB)
      CALL SCHDL(4,1000.,ATRIB)
      RETURN
      END
```

```
C ****************************************
C          THE POLLING SUBROUTINE
C ****************************************
 SUBROUTINE POLL
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
1,ELMTP,ELMTR
 DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
 ATRIB(1)=TNOW
 IF(POLC.EQ.1) RETURN
1000 IF(NC.EQ.K) GO TO 3333
      CALL NEXT(POLC)
      IF(POLC.EQ.1) RETURN
      IF(POLC.EQ.2) RETURN
      IF(JET.EQ.1) GO TO 9000
249  CALL RNG(IRAND)
 C    IF(IRAND.LT.ELMTP) GO TO 200
      IF((IRAND.GE.1).AND.(IRAND.LE.ELMTP)) GO TO 200
      GO TO 250
200   ERR(M)=ERR(M)+1
      IF(ERR(M).EQ.LIMIT) GO TO 201
      GO TO 214
201   DST(JS)=M
      JS=JS+1
214   DG=13.+3.
      CALL SCHDL(3,DG,ATRIB)
      CALL SCHDL(1,DG,ATRIB)
      RETURN
250  IF(RCL(M).GT.0.0) GO TO 300
     GO TO 350
300  INT=RCL(M)
     CALL SCHDL(8,3.,ATRIB)
     RETURN
350  CALL SCHDL(9,3.,ATRIB)
     RETURN
```

```
9000 JET=0.0
     CALL SCHDL(1,TOL,ATRIB)
     RETURN
3333 TOSS=NC
     CALL COLCT(TOSS,3)
     RETURN
     END
C ************************************************************
C                       SUBROUTINE  REMOVAL
C THIS SUBROUTINE IS USED TO REMOVE THE RTU DATA MSG'S WHICH
C HAVE BEEN ACKNOWLEDGED PREVIOUSLY FROM THE MASTER STATION
C RCL(I) > 0. IT ALSO SCHEDULES THE END OF SERVICE FOR THE
C RTU DATA MSG'S AFTER CALLING SUBROUTINE RNGR.
C ************************************************************
      SUBROUTINE REMOVAL
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
     1,TNEXT,TNOW,XX(100)
      COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
     1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
     1,ELMTP,ELMTR
      DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
      ATRIB(1)=TNOW
      DO 311 I=1,INT
         DO 310 J=1,1
         CALL RMOVE(J,M,DUMP)
310      CONTINUE
311   CONTINUE
      RCL(M)=0.0
      DF=NNQ(M)
      IDF=DF
      SQ=DF*13.
      CALL SCHDL(7,SQ,ATRIB)
      DO 320 J=1,IDF
      CALL RNGR(IRANDR)
C     IF(IRANDR.LT.ELMTR) GO TO 330
      IF((IRANDR.GE.100).AND.(IRANDR.LE.ELMTR)) GO TO 330
      RCL(M)=RCL(M)+1
      ERR(M)=NNQ(M)-RCL(M)
```

```
320   CONTINUE
       GO TO 340
330   ERR(M)=NNQ(M)-RCL(M)
      IF(ERR(M).EQ.LIMIT) GO TO 303
      GO TO 304
303   DST(JS)=M
      JS=JS+1
304   DQ=DF*13.
      CALL SCHDL(1,DQ,ATRIB)
      RETURN
340   ZQ=DF*13.
      CALL SCHDL(1,ZQ,ATRIB)
      RETURN
      END
C ************************************************************
C                      THE SAVE SUBROUTINE
C THIS SUBROUTINE IS USED TO SCHEDULE THE END OF SERVICE FOR
C THE RTU DATA MSG'S WHEN RCL(I) < 0. IT ALSO CALLS
C SUBROUTINE RNGR TO CHECK WHETHER THE RTU DATA MSG'S ARE
C RECEIVED CORRECTLY OR NOT.
C *************************************************************
 SUBROUTINE SAVE
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
1,ELMTP,ELMTR
 DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  DF=NNQ(M)
  IDF=DF
  DR=DF*13.
      CALL SCHDL(6,DR,ATRIB)
      DO 720 J=1,IDF
        CALL RNGR(IRANDR)
C       IF(IRANDR.LT.ELMTR) GO TO 730
        IF((IRANDR.GE.100).AND.(IRANDR.LE.ELMTR)) GO TO 730
        RCL(M)=RCL(M)+1
```

```
        ERR(M)=NNQ(M)-RCL(M)
720   CONTINUE
      GO TO 740
730   ERR(M)=NNQ(M)-RCL(M)
      IF(ERR(M).EQ.LIMIT) GO TO 703
      GO TO 704
703   DST(JS)=M
      JS=JS+1
704   DQ=DF*13.
      CALL SCHDL(1,DQ,ATRIB)
      RETURN
740   ZQ=DF*13.
      CALL SCHDL(1,ZQ,ATRIB)
      RETURN
      END
C ***********************************************************
C                         SUBROUTINE NEXT
C THIS SUBROUTINE IS USED TO POLL THE NEXT RTU AND MAKES
C SURE THAT BEFORE POLLING ANY RTU IT IS NOT REMOVED FROM
C THE POLLING CYCLE.
C ************************************************************
 SUBROUTINE NEXT(POLC)
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
1,ELMTP,ELMTR
 DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
      ATRIB(1)=TNOW
      POLC=0.0
      A=M
      M=M+1
      IF(A.EQ.(MIN-1)) GO TO 540
      IF(M.EQ.(MAX+1)) GO TO 500
      GO TO 540
500   JET=1.0
      M=MIN-1
C SET A POLLING CYCLE DELAY
```

```
      TOL=((NC*1000)-TNOW)
      CALL COLCT(TOL,1)
      IF(TOL.LT.0) GO TO 518
      NC=NC+1
      RETURN
540   DO 510 J=1,150
      IF(M.EQ.DST(J)) GO TO 515
510   CONTINUE
      GO TO 520
515   M=M+1
      IF((A.EQ.(MIN-1)).AND.(M.EQ.(MAX+1))) GO TO 517
      IF(M.EQ.(MAX+1)) GO TO 530
      GO TO 540
530   JET=1.0
      M=MIN-1
      TOL=((NC*1000)-TNOW)
      CALL COLCT(TOL,1)
      IF(TOL.LT.0) GO TO 518
      NC=NC+1
      RETURN
520   DS=3.0
      CALL SCHDL(5,DS,ATRIB)
      GO TO 535
518   POLC=2
      CALL COLCT(POLC,2)
      GO TO 535
517   POLC=1
      CALL COLCT(POLC,2)
535   RETURN
      END
C ***********************************************************
C                          SUBROUTINE RNG
C THIS SUBROUTINE GENERATES (2**22)-1 RANDOM NMUBERS.
C ISR(I): IS AN ARRAY USED TO STORE THE VALUES OF THE SHIFT
C REGISTERS. THIS RNG HAS 22 SHIFT REG'S AND CAN GENERATE A
C MAXIMUM OF (2**22)-1 = 4,194,303 RANDOM NUMBERS WHICH ARE
C UNIFORMLY DISTRIBUTED.
C
C IFEED : IS THE FEEDBACK CONNECTION TAKEN FROM SR(22) &
```

```
C SR(1) AND THEN USED AS THE NEW VALUE FOR SR(1).
C IRAND : IS THE OUTPUT RANDOM NUMBER.
C ***********************************************************
  SUBROUTINE RNG(IRAND)
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
       ATRIB(1)=TNOW
       IFEED=(ISR(22)+ISR(1))
       IF(IFEED.EQ.2) IFEED=0
           K=22
       DO 900 I=2,22
             ISR(K)=ISR(K-1)
             K=K-1
900    CONTINUE
 ISR(1)=IFEED
 IRD1=(ISR(22)*1.D00)+(ISR(21)*2.D00)+(ISR(20)*4.D00)
 1+(ISR(19)*8.D00)+(ISR(18)*16.D00)+(ISR(17)*32.D00)+
 1(ISR(16)*64.D00)+(ISR(15)*128.D00)
 IRD2=(ISR(14)*(2**8))+(ISR(13)*(2**9))+(ISR(12)*(2**10))
1+(ISR(11)*(2**11))+(ISR(10)*(2**12))+(ISR(9)*(2**13))
1+(ISR(8)*(2**14))
 IRD3=(ISR(7)*32768.D00)+(ISR(6)*65536.D00)+(ISR(5)
1*131072.D00)+(ISR(4)*262144.D00)+(ISR(3)*524288.D00)+
1(ISR(2)*1048576.D00)+(ISR(1)*2097152.D00)
 IRAND=IRD1+IRD2+IRD3
 RETURN
 END
C ***********************************************************
C                        SUBROUTINE RNGR
C THIS SUBROUTINE GENERATES (2**23)-1 RANDOM NMUBERS.
C IS(I): IS AN ARRAY USED TO STORE THE VALUES OF THE SHIFT
C REGISTERS. THIS RNG HAS 22 SHIFT REG'S AND CAN GENERATE A
C MAXIMUM OF (2**23)-1 = 8,388,607 RANDOM NUMBERS WHICH ARE
C UNIFORMLY DISTRIBUTED.
```

```
 C IFED : IS THE FEEDBACK CONNECTION TAKEN FROM IS(23) &
 C IS(5) AND THEN USED AS THE NEW VALUE FOR IS(1).
 C IRANDR: IS THE OUTPUT RANDOM NUMBER.
 C ***********************************************************
 SUBROUTINE RNGR(IRANDR)
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
       ATRIB(1)=TNOW
      IFED=(IS(23)+IS(5))
      IF(IFED.EQ.2) IFED=0
      K=23

           DO 833 I=2,23
               IS(K)=IS(K-1)
               K=K-1
833        CONTINUE
 IS(1)=IFED
 IR1=(IS(23)*1.D00)+(IS(22)*2.D00)+(IS(21)*4.D00)+(IS(20)
 1*8.D00)+(IS(19)*16.D00)+(IS(18)*32.D00)+(IS(17)*64.D00)
 1+(IS(16)*128.D00)+(IS(15)*256.D00)
 IR2=(IS(14)*(2**9))+(IS(13)*(2**10))+(IS(12)*(2**11))+
 1(IS(11)*(2**12))+(IS(10)*(2**13))+(IS(9)*(2**14))+(IS(8)
 1*(2**15))
IR3=(IS(7)*65536.D00)+(IS(6)*131072.D00)+(IS(5)*262144.D00)+
1(IS(4)*524288.D00)+(IS(3)*1048576.D00)+(IS(2)*2097152.D00)
1+(IS(1)*4194304.D00)
 IRANDR=IR1+IR2+IR3
 RETURN
 END
C ***********************************************************
C                       THE INPUT SUBROUTINE
C THIS SUBROUTINE IS USED TO INPUT ONE DATA MSG TO EACH RTU
C WHICH HAS NOT BEEN REMOVED FROM THE POLLING CYCLE.
C ***********************************************************
```

```
 SUBROUTINE INPUT
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
1,ELMTP,ELMTR
 DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
 ATRIB(1)=TNOW
    DO 800 J=MIN,MAX
         DO 8000 K=1,150
            IF(DST(K).EQ.J) GO TO 800
8000        CONTINUE
       CALL FILEM(J,1)
800   CONTINUE
      CALL SCHDL(4,1000.,ATRIB)
      RETURN
 END
C ********************************************************
C                     THE RELAX SUBROUTINE
C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES.
C ********************************************************
 SUBROUTINE RELAX
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
C ********************************************************
C                     THE ENDSV SUBROUTINE
C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES.
C ********************************************************
 SUBROUTINE ENDSV
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
```

```
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
 C *****************************************************
 C                  THE TEST SUBROUTINE
 C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES.
 C *****************************************************
 SUBROUTINE TEST
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
 C *****************************************************
 C                  THE HELP SUBROUTINE
 C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES.
 C *****************************************************
 SUBROUTINE HELP
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
```

```
C *********************************************************
C                 THE TRIAL SUBROUTINE
C THIS SUBROUTINE IS WRITTEN FOR TROUBLESHHOTING PURPOSES.
C *********************************************************
 SUBROUTINE TRIAL
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
C *****************************************************
C                THE OUTPUT SUBROUTINE
C  THIS SUBROUTINE IS USED TO PRINT THE OUTPUT REPORT
C *****************************************************
  SUBROUTINE OTPUT
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  Z=0.0
  WT=0.0
       DO 55 J=MIN,MAX
             Z=FFAWT(J)+Z
   55  CONTINUE
   Z=Z/((MAX-MIN)+1)
   WRITE(*,*) Z
   CALL COLCT(Z,8)
        DO 65 J=1,50
              WT=DST(J)
              IF(WT.GT.0.0) GO TO 75
              GO TO 65
```

```
75              CALL COLCT(WT,9)
65        CONTINUE
WRITE(*,*)  (DST(J),J=1,50)
RETURN
END
```

APPENDIX C

SLAM II SIMULATION PROGRAM FOR

DATA ACQUISITION & CONTROL

M.S. THESIS

RADIO FREQUENCY SPREAD SPECTRUM

FOR

DISTRIBUTED CONTROL SYSTEM

Done By : **G. B. AL-DANDAN**

THIS PROGRAM IS SIMILAR TO THE PROGRAM WRITTEN IN APPENDIX B WITH SOME MINOR DIFFERENCES. THE MASTER STATION, IN THIS CASE IS ALLOWED TO TRANSMIT CONTROL COMMANDS TO THE VARIOUS RTUS (IN ADDITION TO DATA ACQUISITION).

VARIABLES DEFINITION

ERR(M) : IS THE NUMBER OF DATA MESSAGES THAT ARE LOCATED IN THE M'TH RTU AND HAVE NOT BEEN TRANSMITTED TO THE MASTER STATION DUE TO TRANSMISSION ERROR CAUSED BY EITHER WRONG POLLING MSG OR WRONG RECEPTION OF RTU DATA MSG.

RCL(M) : IT CONTAINS THE NUMBER OF MSG'S THAT SHOULD BE REMOVED FROM THE M'TH RTU WHEN THE CORRECT POLLING MSG REACHES THAT PARTICULAR RTU.

NNQ(M) : IS THE TOTAL NUMBER OF MSG'S AVAILABLE AT THE M'TH RTU AND AT THE CURRENT TIME .

DST(JS) : IT CONTAINS THE NUMBER OF RTUS WHICH HAVE BEEN REMOVED FROM THE POLLING CYCLE.

TOL : IS THE TIME WHICH THE MASTER STATION SHOULD WAIT AT THE END OF EACH POLLING CYCLE (TOLERANCE).

RTN : THE TOTAL NUMBER OF RTUS IN THE NETWORK.

JET	:	1 INDICATES MASTER STATION SHOULD STOP POLLING THE RTUS AND WAITS A TIME SPECIFIED BY " TOL " VALUE.
POLC	:	1 INDICATES THAT ALL RTUS HAVE BEEN REMOVED FROM THE POLLING CYCLE. 2 INDICATES THAT THE TOLERANCE IS NEGATIVE.
LIMIT	:	IS THE MAXIMUM NUMBER OF ERROR MSG'S ALLOWED BEFORE A PARTICULAR RTU IS REMOVED FROM POLLING CYCLE.
ELMTP	:	A CERTAIN VALUE IF EXCEEDED THE POLLING MSG WILL BE IN ERROR.
ELMTPC	:	A CERTAIN VALUE IF EXCEEDED THE POLLING MSG (CONTROL) WILL BE IN ERROR
ELMTR	:	A CERTAIN VALUE IF EXCEEDED THE RTU DATA MSG WILL BE IN ERROR.
PER	:	PACKET P_e FOR THE RTU DATA MSG.
SP	:	IS THE DATA RATE USED.

RCOM(M) : IS AN ARRAY CONTAINING THE NUMBER OF THE RTUs WHICH CAN TRANSMIT BOTH DATA AND CONTROL MSGs.

INRC : TOTAL NUMBER OF RTUs WHICH ACCEPT CONTROL COMMANDS.

MAX : IS THE MAXIMUM NUMBER OF RTUs USED.

```
$STORAGE:2
$NOTSTRICT
$PAGESIZE:60
$NOTLARGE
$NOFLOATCALLS
 PROGRAM MAIN
 DIMENSION NSET(20000)
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
 COMMON QSET(20000)
 DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
 EQUIVALENCE(NSET(1),QSET(1))
 NNSET=20000
 NCRDR=5
 NPRNT=6
 NTAPE=7
 CALL SLAM
 STOP ' '
 END
C ****************************************
C       THE EVEVT SUBROUTINE
C ****************************************
 SUBROUTINE EVENT(I)
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 GO TO (1,2,3,4,5,6,7,8,9) ,I
1    CALL POLL
     RETURN
2    CALL RELAX
     RETURN
3    CALL ENDSV
     RETURN
4    CALL INPUT
     RETURN
```

```
5     CALL TEST
      RETURN
6     CALL HELP
      RETURN
7     CALL TRIAL
      RETURN
8     CALL REMOVAL
      RETURN
9     CALL SAVE
      RETURN
      END
C ****************************************
C       THE INITIALIZATION SUBROUTINE
C ****************************************
  SUBROUTINE INTLC
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  PER=
  ELMTP=
  ELMTPC=
  ELMTR=
  SP=445000
  LIMIT=4
  MAX=42
  MIN=3
  M=MIN-1
  JS=1
  POLC=0
  TOL=0
  JET=0.0
  NC=1
  INT=1
C INITIAL REGISTER VALUES OF THE RANDOM NUMBER GENERATOR
C (IRAND).
```

```
      ISR(1)=0
      ISR(2)=1
      ISR(3)=0
      ISR(4)=0
      ISR(5)=0
      ISR(6)=0
      ISR(7)=0
      ISR(8)=0
      ISR(9)=0
      ISR(10)=0
      ISR(11)=0
      ISR(12)=0
      ISR(13)=0
      ISR(14)=1
      ISR(15)=0
      ISR(16)=0
      ISR(17)=0
      ISR(18)=0
      ISR(19)=0
      ISR(20)=0
      ISR(21)=0
      ISR(22)=0
C INITIAL REGISTER VALUES OF THE RANDOM NUMBER GENERATOR
C (IRANDR).
      IS(1)=0
      IS(2)=0
      IS(3)=0
      IS(4)=0
      IS(5)=0
      IS(6)=0
      IS(7)=0
      IS(8)=0
      IS(9)=0
      IS(10)=0
      IS(11)=0
      IS(12)=0
      IS(13)=0
      IS(14)=0
      IS(15)=0
```

```
 IS(16)=0
 IS(17)=1
 IS(18)=0
 IS(19)=0
 IS(20)=0
 IS(21)=0
 IS(22)=0
 IS(23)=0
C INITIALIZATION  OF RCOM(I)
 RCOM(1)= 3
 RCOM(2)= 4
 RCOM(3)= 5
 RCOM(4)= 6
 RCOM(5)= 7
 RCOM(6)= 8
 RCOM(7)= 9
 RCOM(8)= 10
 RCOM(9)= 11
 RCOM(10)=12
 RCOM(11)=13
 RCOM(12)=14
 RCOM(13)=15
 RCOM(14)=16
 RCOM(15)=17
 RCOM(16)=18
 RCOM(17)=19
 RCOM(18)=20
 RCOM(19)=21
 RCOM(20)=22
 RCOM(21)=23
 RCOM(22)=24
 DO 10 J=1,150
       ERR(J)=0
       RCL(J)=0
       DST(J)=0
 10  CONTINUE
 TL=LIMIT
 CALL COLCT(TL,4)
 CALL COLCT(PER,5)
```

```
 RTN=(MAX-MIN)+1
 INRC=RTN*0.5
 PTO=INRC
 CALL COLCT(RTN,6)
 CALL COLCT(SP,7)
 CALL COLCT(PTO,11)
C PUT ONE MSG IN EVERY RTU FILE AT T=0.0
 DO 30 I=1,1
   DO 20 J=MIN,MAX
       CALL FILEM(J,I)
20    CONTINUE
30   CONTINUE
C      SCHEDULE EVENT 1 TO START AT TIME=0.0 AND INPUT DATA
C      MESSAGES TO THE RTU'S USING EVENT 4 EVERY 999 TIME
C      UNITS
       CALL SCHDL(1,0.0,ATRIB)
       CALL SCHDL(4,1000.,ATRIB)
       RETURN
       END
C ****************************************
C        THE POLLING SUBROUTINE
C ****************************************
 SUBROUTINE POLL
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
 DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
 ATRIB(1)=TNOW
      DO 3311 J=MIN,MAX
           IF(NNQ(J).GT.9) GO TO 807
      3311 CONTINUE
      IF(POLC.EQ.1) RETURN
1000 IF(NC.EQ.K) GO TO 3333
      CALL NEXT(POLC)
      IF(POLC.EQ.1) RETURN
      IF(POLC.EQ.2) RETURN
```

```
      IF(JET.EQ.1) GO TO 9000
 DO 616 I=1,INRC
      IF(M.EQ.RCOM(I)) GO TO 2147
616   CONTINUE
      GO TO 249
2147   CALL RNG(IRAND)
       IF((IRAND.GE.1).AND.(IRAND.LE.ELMTPC)) GO TO 200
       GO TO 250
249   CALL RNG(IRAND)
C      IF(IRAND.LT.ELMTP) GO TO 200
       IF((IRAND.GE.1).AND.(IRAND.LE.ELMTP)) GO TO 200
       GO TO 250
200    ERR(M)=ERR(M)+1
       IF(ERR(M).EQ.LIMIT) GO TO 201
       GO TO 214
201    DST(JS)=M
       JS=JS+1
214    DO 611 I=1,INRC
       IF(M.EQ.RCOM(I)) GO TO 2141
611    CONTINUE
       DG=13.+3.
       GO TO 2142
2141   DG=13.+7.
2142   CALL SCHDL(3,DG,ATRIB)
       CALL SCHDL(1,DG,ATRIB)
       RETURN
C CORRECT POLLING MESSAGE
250   IF(RCL(M).GT.0.0) GO TO 300
      GO TO 350
300   INT=RCL(M)
      DO 612 I=1,INRC
           IF(M.EQ.RCOM(I)) GO TO 2143
612   CONTINUE
      CALL SCHDL(8,3.,ATRIB)
      RETURN
2143 CALL SCHDL(8,7.,ATRIB)
      RETURN
350   DO 613 I=1,INRC
      IF(M.EQ.RCOM(I)) GO TO 2144
```

```
 613   CONTINUE
       CALL SCHDL(9,3.,ATRIB)
       RETURN
 2144  CALL SCHDL(9,7.,ATRIB)
       RETURN
 9000  JET=0.0
       CALL SCHDL(1,TOL,ATRIB)
       RETURN
 807   WRN=1
       CALL COLCT(WRN,10)
       RETURN
 3333  TOSS=NC
       CALL COLCT(TOSS,3)
       RETURN
       END
C ********************************************************
C                  SUBROUTINE  REMOVAL
C THIS SUBROUTINE IS USED TO REMOVE THE RTU DATA MSG'S WHICH
C HAVE BEEN ACKNOWLEDGED PREVIOUSLY FROM THE MASTER STATION
C RCL(I) > 0. IT ALSO SCHEDULES THE END OF SERVICE FOR THE
C RTU DATA MSG'S AFTERC CALLING SUBROUTINE RNGR.
C ********************************************************
  SUBROUTINE REMOVAL
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
       DO 311 I=1,INT
            DO 310 J=1,1
                CALL RMOVE(J,M,DUMP)
       310  CONTINUE
 311   CONTINUE
 RCL(M)=0.0
 DF=NNQ(M)
 IDF=DF
```

```
 SQ=DF*13.
 CALL SCHDL(7,SQ,ATRIB)
 DO 320 J=1,IDF
      CALL RNGR(IRANDR)
C     IF(IRANDR.LT.ELMTR) GO TO 330
      IF((IRANDR.GE.100).AND.(IRANDR.LE.ELMTR)) GO TO 330
      RCL(M)=RCL(M)+1
      ERR(M)=NNQ(M)-RCL(M)
320  CONTINUE
 GO TO 340
330  ERR(M)=NNQ(M)-RCL(M)
IF(ERR(M).EQ.LIMIT) GO TO 303
 GO TO 304
303  DST(JS)=M
     JS=JS+1
304  DQ=DF*13.
 CALL SCHDL(1,DQ,ATRIB)
 RETURN
340  ZQ=DF*13.
 CALL SCHDL(1,ZQ,ATRIB)
 RETURN
 END
C ************************************************************
C                  THE SAVE SUBROUTINE
C THIS SUBROUTINE IS USED TO SCHEDULE THE END OF SERVICE FOR
C THE RTU DATA MSG'S WHEN RCL(I) < 0. IT ALSO CALLS
C SUBROUTINEC RNGR TO CHECK WHETHER THE RTU DATA MSG'S ARE
C RECEIVED CORRECTLYC OR NOT.
C ************************************************************
 SUBROUTINE SAVE
 COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
 COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
 DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
 ATRIB(1)=TNOW
 DF=NNQ(M)
```

```
  IDF=DF
  DR=DF*13.
  CALL SCHDL(6,DR,ATRIB)
  DO 720 J=1,IDF
        CALL RNGR(IRANDR)
C       IF(IRANDR.LT.ELMTR) GO TO 730
        IF((IRANDR.GE.100).AND.(IRANDR.LE.ELMTR)) GO TO 730
        RCL(M)=RCL(M)+1
        ERR(M)=NNQ(M)-RCL(M)
720  CONTINUE
 GO TO 740
730  ERR(M)=NNQ(M)-RCL(M)
  IF(ERR(M).EQ.LIMIT) GO TO 703
  GO TO 704
703  DST(JS)=M
 JS=JS+1
704  DQ=DF*13.
 CALL SCHDL(1,DQ,ATRIB)
 RETURN
740  ZQ=DF*13.
  CALL SCHDL(1,ZQ,ATRIB)
  RETURN
  END
C ***********************************************************
C                     SUBROUTINE NEXT
C THIS SUBROUTINE IS USED TO POLL THE NEXT RTU AND MAKES
C SURE THAT BEFORE POLLING ANY RTU IT IS NOT REMOVED FROM
C THE POLLING CYCLE.
C ************************************************************
  SUBROUTINE NEXT(POLC)
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  POLC=0.0
```

```
 A=M
 M=M+1
 IF(A.EQ.(MIN-1)) GO TO 540
 IF(M.EQ.(MAX+1)) GO TO 500
 GO TO 540
500  JET=1.0
     M=MIN-1
C SET A POLLING CYCLE DELAY
 TOL=((NC*1000)-TNOW)
 CALL COLCT(TOL,1)
 IF(TOL.LT.0) GO TO 518
 NC=NC+1
 RETURN
540  DO 510 J=1,150
     IF(M.EQ.DST(J)) GO TO 515
510  CONTINUE
     GO TO 520
515  M=M+1
     IF((A.EQ.(MIN-1)).AND.(M.EQ.(MAX+1))) GO TO 517
     IF(M.EQ.(MAX+1)) GO TO 530
     GO TO 540
530  JET=1.0
     M=MIN-1
     TOL=((NC*1000)-TNOW)
     CALL COLCT(TOL,1)
     IF(TOL.LT.0) GO TO 518
     NC=NC+1
     RETURN
520  DO 614 I=1,INRC
     IF(M.EQ.RCOM(I)) GO TO 2145
614  CONTINUE
     DS=3.0
     GO TO 2147
2145 DS=7.0
2147 CALL SCHDL(5,DS,ATRIB)
     GO TO 535
518  POLC=2
     CALL COLCT(POLC,2)
     GO TO 535
```

```
 517   POLC=1
       CALL COLCT(POLC,2)
 535   RETURN
       END
C ***********************************************************
C                        SUBROUTINE RNG
C THIS SUBROUTINE GENERATES (2**22)-1 RANDOM NMUBERS.
C ISR(I): IS AN ARRAY USED TO STORE THE VALUES OF THE SHIFT
C REGISTERS.THIS RNG HAS 22 SHIFT REG'S AND CAN GENERATE A
C MAXIMUM OF (2**22)-1 = 4,194,303 RANDOM NUMBERS WHICH ARE
C UNIFORMLY DISTRIBUTED.
C IFEED : IS THE FEEDBACK CONNECTION TAKEN FROM SR(22) &
C SR(1) AND THEN USED AS THE NEW VALUE FOR SR(1).
C IRAND : IS THE OUTPUT RANDOM NUMBER.
C ************************************************************
      SUBROUTINE RNG(IRAND)
      COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
     1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
     1,TNEXT,TNOW,XX(100)
      COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
     1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
     1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
      DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
      ATRIB(1)=TNOW
      IFEED=(ISR(22)+ISR(1))
      IF(IFEED.EQ.2) IFEED=0
          K=22
          DO 900 I=2,22
             ISR(K)=ISR(K-1)
             K=K-1
 900      CONTINUE
      ISR(1)=IFEED
      IRD1=(ISR(22)*1.D00)+(ISR(21)*2.D00)+(ISR(20)*4.D00)+
     1(ISR(19)*8.D00)+(ISR(18)*16.D00)+(ISR(17)*32.D00)+(ISR(16)
     1*64.D00)+(ISR(15)*128.D00)
      IRD2=(ISR(14)*(2**8))+(ISR(13)*(2**9))+(ISR(12)*(2**10))
     1+(ISR(11)*(2**11))+(ISR(10)*(2**12))+(ISR(9)*(2**13))
     1+(ISR(8)*(2**14))
      IRD3=(ISR(7)*32768.D00)+(ISR(6)*65536.D00)+(ISR(5)
```

```
1*131072.D00)+(ISR(4)*262144.D00)+(ISR(3)*524288.D00)+
1(ISR(2)*1048576.D00)+(ISR(1)*2097152.D00)
  IRAND=IRD1+IRD2+IRD3
  RETURN
  END
C *******************************************************
C                         SUBROUTINE RNGR
C THIS SUBROUTINE GENERATES (2**23)-1 RANDOM NMUBERS.
C IS(I): IS AN ARRAY USED TO STORE THE VALUES OF THE SHIFT
C REGISTERS.THIS RNG HAS 22 SHIFT REG'S AND CAN GENERATE A
C MAXIMUM OF (2**23)-1 = 8,388,607 RANDOM NUMBERS WHICH ARE
C UNIFORMLY DISTRIBUTED.
C IFED : IS THE FEEDBACK CONNECTION TAKEN FROM IS(23) &
C IS(5) AND THEN USED AS THE NEW VALUE FOR IS(1).
C IRANDR: IS THE OUTPUT RANDOM NUMBER.
C ********************************************************
  SUBROUTINE RNGR(IRANDR)
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  IFED=(IS(23)+IS(5))
  IF(IFED.EQ.2) IFED=0
      K=23
      DO 833 I=2,23
        IS(K)=IS(K-1)
        K=K-1
 833  CONTINUE
  IS(1)=IFED
  IR1=(IS(23)*1.D00)+(IS(22)*2.D00)+(IS(21)*4.D00)+(IS(20)
1*8.D00)+(IS(19)*16.D00)+(IS(18)*32.D00)+(IS(17)*64.D00)+
1(IS(16)*128.D00)+(IS(15)*256.D00)
  IR2=(IS(14)*(2**9))+(IS(13)*(2**10))+(IS(12)*(2**11))+
1(IS(11)*(2**12))+(IS(10)*(2**13))+(IS(9)*(2**14))+(IS(8)
1*(2**15))
```

```
 IR3=(IS(7)*65536.D00)+(IS(6)*131072.D00)+(IS(5)*262144.D00)+
 1(IS(4)*524288.D00)+(IS(3)*1048576.D00)+(IS(2)*2097152.D00)
 1+(IS(1)*4194304.D00)
  IRANDR=IR1+IR2+IR3
  RETURN
  END
C ***********************************************************
C                  THE INPUT SUBROUTINE
C THIS SUBROUTINE IS USED TO INPUT ONE DATA MSG TO EACH RTU
C WHICH HAS NOT BEEN REMOVED FROM THE POLLING CYCLE.
C ***********************************************************
  SUBROUTINE INPUT
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  DO 800 J=MIN,MAX
       DO 8000 K=1,150
          IF(DST(K).EQ.J) GO TO 800
8000   CONTINUE
       CALL FILEM(J,1)
800   CONTINUE
 CALL SCHDL(4,1000.,ATRIB)
 RETURN
 END
C *********************************************************
C                THE RELAX SUBROUTINE
C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES
C *********************************************************
  SUBROUTINE RELAX
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
```

```
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
C ********************************************************
C                      THE ENDSV SUBROUTINE
C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES
C ********************************************************
  SUBROUTINE ENDSV
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
C ********************************************************
C                      THE TEST SUBROUTINE
C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES
C ********************************************************
  SUBROUTINE TEST
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
```

```
C ******************************************************
C                   THE HELP SUBROUTINE
C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES
C *******************************************************
  SUBROUTINE HELP
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
C ******************************************************
C                   THE TRIAL SUBROUTINE
C  THIS SUBROUTINE IS WRITTEN FOR TROUBLESHOOTING PURPOSES
C *******************************************************
  SUBROUTINE TRIAL
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  ATRIB(1)=TNOW
  RETURN
  END
C *******************************************************
C               THE OUTPUT SUBROUTINE
C  THIS SUBROUTINE IS WRITTEN TO OUTPUT THE VALUES OF SOME
C  VARIABLES  WHICH ARE SIGNIFICIENT FOR OUR NETWORK
C  SIMULATION.
C ********************************************************
  SUBROUTINE OTPUT
  COMMON/SCOM1/ ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,
1MSTOP,NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100)
```

```
1,TNEXT,TNOW,XX(100)
  COMMON/UCOM1/ M,ISR(30),JS,NC,MAX,MIN,LIMIT,TOL,RCL(150)
 1,ERR(150),DST(150),JET,IS(30),IQ,IP,ITR(30),IT(30),INT,SP
 1,ELMTP,ELMTR,RCOM(100),INRC,ELMTPC
  DOUBLE PRECISION IRAND,IRD1,IRD2,IRD3,IRANDR,IR1,IR2,IR3
  Z=0.0
  S=0.0
  WT=0.0
      DO 55 J=MIN,MAX
           Z=FFAWT(J)+Z
           S=FFAVG(J)+S
 55   CONTINUE
      Z=Z/((MAX-MIN)+1)
      S=S/((MAX-MIN)+1)
      WRITE(*,*) Z,S
      CALL COLCT(Z,8)
  DO 65 J=1,50
     WT=DST(J)
     IF(WT.GT.0.0) GO TO 75
     GO TO 65
 75  CALL COLCT(WT,9)
 65  CONTINUE
  WRITE(*,*) (DST(J),J=1,50)
  RETURN
  END
```

REFERENCES

[1] P. Pleinevanx and J.D. Decotiqnie. "Time Critical Communication Networks : Field Buses," IEEE Network, Vol.2 No.3, May, 1988.

[2] Dr. Chiristian Gatar, "Fields Bus," Measurement and Control Vol. 21, March 1988.

[3] N. Harverty, "MIL-STD 1553 - A Standard For Data Communications," Vol.10 No.1, pp 29-33, Jan. 1986.

[4] J.P. Thomesse et al., "An Industrial Instrumentation Local Area Network," IECON 86, Milwaukee, WI, Sept. 1986.

[5] INTEL Corp., "The Bitbus Interconnect Serial Control Bus Specifications," in Distributed Control Modules, 1984.

[6] Instrument Society Of America, "PROWAY LAN Industrial Data Highway," ISA - S72, 01 - 1985.

[7] Bosch, Moeller, Siemens,"PROFIBUS : Process Field Bus," Draft Standard 3.0, May 20, 1987.

[8] R.C. Dixon, "Spread Spectrum Signals," study guide, John Wiley and Sons, 1985.

[9] D. Newman Jr. "FCC Authorizes Spread Spectrum," IEEE Comm. Mag, pp 46-47, July, 1986.

[10] W.C. Fifer and F.J. Bruno. "The Low Cost Packet Radio," Proc. of IEEE, Jan. 1987.

[11] M.B. Pursely, "The Role of Spread Spectrum in Packet Radio Networks," Proc. of IEEE, Jan. 1987.

[12] G.R. Copper and R.W. Netteton, "A Spread Spectrum Technique For High Capacity Mobile Communications," IEEE Trans on Veh. Tech. Nov. 1978.

[13] P. Perret, "Application of Spread Spectrum Radio to Wireless Terminal Communications," NTC, Dec 1980.

[14] M. Kavehrad and P. Mclane. "Performance of Low Complexity Channel Coding and Diversity for Spread Spectrum in Indoor Wireless Communications," BSTJ. pp 1927-1965 Oct. 1985.

[15] K. Pahlavan, "Spread Spectrum for Wireless Local Networks," Proc. IEEE, PCC April 1987.

[16] K. Pahlavan, "Wireless Communications for Office Information Networks," IEEE comm. May 1985.

[17] M. Kavehrad and P.J. Mclane, "Spread Spectrum for Indoor Digital Radio," IEEE Comm. June 1987.

[18] NEC Corporation, "Spread Spectrum Modem - Small LAN System," Computer and Communications, Technical bro-chure, Japan, 1989.

[19] K. Dostert, "Multi-channel Transmission of Measurement and Control Data over Indoor Lines using Spread Spectrum Techniques," ACTA IMEKO 88, ISBN: 1-55617-144-7, pp 151-159, 1988.

[20] R.C. Dixon, "Spread Spectrum Systems," John Wiley and Sons, 1984.

[21] R.C. White,"Experiments with Digital Computer Simulation of Pseudo Random Noise Generators," IEEE Trans. Elect. Comp., June 1967.

[22] R.C. Tansworthe, "Random Number Generated by Linear Recurrence Modulo-Two," Math Computation, Vol 19, April 1965.

[23] D.R. Anderson, "A New Class of Cyclic Codes," SIAM J.APPL. MATH., 16, No 1, 1968.

[24] R. Gold, "Characteristics of Linear Sequences and Their Coset Functions," J. SIAM Appl. Math,14 No.5, Sept. 1966

[25] R. Gold, "Optimal Binary Sequences for Spread Spectrum Multiplexing," IEEE Trans Info. Theory, Oct 1967.

[26] J. Hammond & P. O'Reilly, "Performance Analysis of Local Computer Networks," Addison - Wesley Publishing Co., 1986.

[27] N. Abramson, "The ALOHA System," in Computer Communication Networks, N. Abramson and F. Kuo (eds), Englewood Cliffs, N.J.: Prentice-Hall, 1973.

[28] L. Roberts, "Dynamic Allocation of Satellite Capacity through Packet Reservation," Proc. NCC, pp. 711-716, 1973.

[29] F.A. Tobagi, "Random Access Techniques for Data Transmission over Packet Switched Radio Networks," Ph.D. thesis, Computer Science Dept., UCLA, 1974.

[30] J.L. Capetanakis "Tree Algorithms for Packet Broadcast channels," IEEE Trans. on Information Theory IT-25, pp 505-515, Sept 1979.

[31] L. Kleinrock and Yemini, "An Optimal Adaptive Scheme for Multiple Access Broadcast Communication," Proc. of the ICC, 7.2.1-7.2.5, June 1987.

[32] I. Chlamtac, W.R. Franta and K.D. Levin, "BRAM : The Broadcast Recognizing Access Method," IEEE Trans. on Comms COM-27, pp 1183-1189, Aug. 1979.

[33] A.G. Konheim and B. Meister,"Service in a Loop System," J. Ass. Compting Mach., Vol. 19, pp 92-108, Jan. 1972.

[34] F.A. Tobagi and Kleinrock, "Packet Switching in Radio Channels: Part III- Polling and (dynamic) Split Channel Reservation Multiple Access," IEEE Trans. Comms., Vol Comm-24, pp 832-845, Aug. 1976.

[35] General Motors, report on "Manufacturing Automation Protocol (MAP)," Feb 20, 1985.

[36] M. Kavehrad, and G. Bodeep,"Design and Experimental Results for a Direct Sequence Spread Spectrum Radio Using Differential Phase-Shift keying Modulation for Indoor, Wireless Communications," IEEE J. Select. Areas Commun., vol SAC-5, pp 815-823, June 1987.

[37] K.S. Shanmugam, "Digital and Analog Communication Systems," John Wiley and Sons, 1979.

[38] ANSI/IEEE STD 802.4, "Token - Passing Bus Access Method and Physical Layer Specifications," Feb 25, 1985.

[39] W. Stallings, "Data and Computer Communications," Macmillan Publishing Co., pp 108, 1985.

[40] A. Alan and B. Pritsker, "Introduction to Simulation and SLAM II," John Wiley & Sons, 3rd edition, 1986.

VITA

Ghaleb B. AL-Dandan was born in Beirut, Lebanon in 1964, where he received his high school education. In 1985 he graduated from King Fahd University of Petroleum and Minerals with a B.Sc in EE. Then he joined Saudi ARAMCO where he is currently working as a computer communication network engineer. He completed his master degree in EE from KFUPM in 1989, as a part-time student. His current interest is in the application of FDDI and ISDN technologies. Ghaleb is a member of IEEE since 1985.